IWR14xx/16xx/18xx/68xx/64xx Industrial Radar Family

Technical Reference Manual

Literature Number: SWRU522E
May 2017–Revised June 2020

Contents

Preface **133**

1 14xx **134**

1.1 14xx Introduction 135

1.1.1 14xx Overview 135

1.1.2 14xx Description 135

1.2 14xx Memory Map 138

1.2.1 System Interconnect 138

1.2.2 Master Subsystem Cortex-R4F Memory Map 139

1.2.3 EDMA Memory Map 142

1.3 14xx Integration 143

1.3.1 Cortex-R4F Subsystem 143

1.3.2 Clock Comparator 144

1.3.3 Direct Memory Access Controller (MSS_DMA) 146

1.3.4 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB) 148

1.3.5 General Purpose I/0 (MSS_GIO) 148

1.3.6 Vectored Interrupt Manager (MSS_VIM) 149

1.3.7 Controller Area Network (MSS_DCAN) 152

1.3.8 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPIA) 153

1.3.9 Serial Peripheral Interface (MSS_SPIB) 153

1.3.10 Quad Serial Peripheral Interface (MSS_QSPI) 154

1.3.11 Enhanced Direct Memory Access (EDMA) 154

1.3.12 Error Signaling Module (MSS_ESM) 157

1.3.13 High-Speed Interface (HSI) 160

1.3.14 Handshake RAM (HSRAM) 160

2 16xx/18xx **161**

2.1 16xx Introduction 162

2.1.1 16xx Overview 162

2.1.2 16xx Description 162

2.2 16xx Memory Map 166

2.2.1 System Interconnect 166

2.2.2 Master Subsystem Cortex-R4F Memory Map 167

2.2.3 DSP Subsystem Memory Map 171

2.2.4 EDMA Memory Map 172

2.3 16xx Integration 173

2.3.1 Cortex-R4F Subsystem 173

2.3.2 Clock Comparator 174

2.3.3 C674x DSP Subsystem 176

2.3.4 Direct Memory Access Controller (MSS_DMA) 179

2.3.5 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB) 183

2.3.6 General Purpose I/0 (MSS_GIO) 183

2.3.7 Data Modification Module (MSS_DMM) 185

2.3.8 Enhanced Pulse Width Modulator (MSS_ETPWM) 186

2.3.9 Vectored Interrupt Manager (MSS_VIM) 187

2.3.10 Controller Area Network (MSS_DCAN) 190

2.3.11 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPI) 191

2.3.12 Quad Serial Peripheral Interface (MSS_QSPI) 192
2.3.13 Enhanced Direct Memory Access (EDMA) 192
2.3.14 Error Signaling Module (MSS_ESM/DSS_ESM) 199
2.3.15 High-Speed Interface (HSI) 203
2.3.16 Handshake RAM (DSS_HSRAM1) 203
2.4 18xx Introduction 203
2.4.1 18xx Overview 203
2.4.2 18xx Description 204
2.5 18xx Memory Map 207
2.5.1 System Interconnect 207
2.5.2 Master Subsystem Cortex-R4F Memory Map 208
2.5.3 DSP Subsystem Memory Map 213
2.5.4 EDMA Memory Map 214
2.6 18xx Integration 215
2.6.1 Cortex-R4F Subsystem 215
2.6.2 Clock Comparator 216
2.6.3 C674x DSP Subsystem 218
2.6.4 Direct Memory Access Controller (MSS_DMA) 221
2.6.5 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB) 225
2.6.6 General Purpose I/0 (MSS_GIO) 225
2.6.7 Data Modification Module (MSS_DMM) 227
2.6.8 Enhanced Pulse Width Modulator (MSS_ETPWM) 228
2.6.9 Vectored Interrupt Manager (MSS_VIM) 229
2.6.10 Controller Area Network (MSS_DCAN) 232
2.6.11 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPI) 233
2.6.12 Quad Serial Peripheral Interface (MSS_QSPI) 234
2.6.13 Enhanced Direct Memory Access (EDMA) 234
2.6.14 Error Signaling Module (MSS_ESM/DSS_ESM) 241
2.6.15 High-Speed Interface (HSI) 245
2.6.16 Handshake RAM (DSS_HSRAM1) 245
3 68xx/64xx 246
3.1 68xx/64xx Introduction 247
3.1.1 Overview 247
3.1.2 Description 247
3.2 Memory Map 251
3.2.1 System Interconnect 251
3.2.2 Master Subsystem Cortex-R4F Memory Map 253
3.2.3 DSP Subsystem Memory Map (Applicable Devices Only) 257
3.2.4 EDMA Memory Map 259
3.3 68xx/64xx Integration 260
3.3.1 Cortex-R4F Subsystem 260
3.3.2 Clock Comparator 261
3.3.3 C674x DSP Subsystem (Applicable Devices Only) 263
3.3.4 Direct Memory Access Controller (MSS_DMA) 266
3.3.5 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB) 269
3.3.6 General Purpose I/0 (MSS_GIO) 269
3.3.7 Data Modification Module (MSS_DMM) 271
3.3.8 Enhanced Pulse Width Modulator (MSS_ETPWM) 272
3.3.9 Vectored Interrupt Manager (MSS_VIM) 273
3.3.10 Module Controller Area Network (MSS_MCAN) 276
3.3.11 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPI) 277
3.3.12 Quad Serial Peripheral Interface (MSS_QSPI) 278
3.3.13 Enhanced Direct Memory Access (EDMA) 278

3.3.14 Error Signaling Module (MSS_ESM/DSS_ESM) 285
3.3.15 High-Speed Interface (HSI) 289
3.3.16 Handshake RAM (DSS_HSRAM1) 289
4 Power, Reset, Clock Management and Control Registers (IWR) 290
4.1 PRCM Overview 291
4.2 Digital Clock Domains 291
4.2.1 Clock Overview 291
4.2.2 Device Clock Sources (For Digital Logic) 292
4.2.3 Device Clock Domains 292
4.2.4 Clock Initialization Sequence 293
4.2.5 PMIC Clock 293
4.3 Resets 294
4.3.1 Reset Overview 294
4.3.2 Reset Types and Sources 295
4.3.3 Reset Domains 295
4.3.4 Reset Cause Registers 296
4.3.5 Module Resets 296
4.4 Power Domain 296
4.4.1 Available Device Power Domains 296
4.4.2 Power Domain Control 297
4.5 Boot 300
4.5.1 Bootloader Modes 300
4.5.2 ROM Eclipsing 300
4.6 14xx Control Registers 301
4.6.1 MSS_TOPRCM Registers 302
4.6.2 MSS_RCM Registers 342
4.6.3 MSS_GPCFG_REG Registers 377
4.6.4 DSS_REG Registers 395
4.7 16xx Control Registers 487
4.7.1 MSS_TOPRCM Registers 488
4.7.2 MSS_RCM Registers 534
4.7.3 MSS_GPCFG_REG Registers 577
4.7.4 DSS_REG Registers 613
4.7.5 DSS_REG2 Registers 727
4.8 68xx/64xx Control Registers 786
4.8.1 MSS_TOPRCM Registers 787
4.8.2 MSS_RCM Registers 833
4.8.3 MSS_GPCFG_REG Registers 888
4.8.4 DSS_REG Registers 924
4.8.5 DSS_REG2 Registers 1038
4.9 18xx Control Registers 1096
4.9.1 MSS_TOPRCM Registers 1097
4.9.2 MSS_RCM Registers 1143
4.9.3 MSS_GPCFG_REG Registers 1186
4.9.4 DSS_REG Registers 1222
4.9.5 DSS_REG2 Registers 1336
5 RADAR Subsystem 1395
5.1 Block Diagram 1396
5.1.1 Clock Subsystem 1397
5.1.2 Transmit Subsystem 1398
5.1.3 Receive Subsystem 1399
6 Master Subsystem Cortex-R4F 1400
6.1 Introduction 1401

6.2 Cortex-R4F Subsystem ... 1401
6.2.1 Cortex-R4F Integration ... 1401
6.2.2 Tightly Coupled Memories ... 1401
6.2.3 Memory Module Hardware Initialization ... 1401

7 Direct Memory Access Controller (DMA) ... 1402
7.1 Module Operation ... 1403
7.1.1 Block Diagram ... 1403
7.1.2 Memory Space ... 1403
7.1.3 DMA Data Access ... 1403
7.1.4 Addressing Modes ... 1405
7.1.5 DMA Channel Control Packets ... 1405
7.1.6 Priority Queue ... 1409
7.1.7 Data Packing and Unpacking ... 1411
7.1.8 DMA Request ... 1414
7.1.9 Auto-Initiation ... 1415
7.1.10 Interrupts ... 1415
7.1.11 Debugging ... 1417
7.1.12 Power Management ... 1417
7.1.13 FIFO Buffer ... 1417
7.1.14 Channel Chaining ... 1418
7.1.15 Memory Protection ... 1419
7.1.16 Parity Checking ... 1420
7.1.17 Parity Testing ... 1421
7.1.18 Initializing RAM with Parity ... 1421
7.2 Control Registers and Control Packets ... 1422
7.2.1 DMA Control Registers ... 1423
7.2.2 Channel Configuration ... 1503
7.3 DMA Request Map ... 1508

8 Vectored Interrupt Manager (VIM) ... 1509
8.1 Overview ... 1510
8.2 Device Level Interrupt Management ... 1511
8.2.1 Interrupt Generation at the Peripheral ... 1511
8.2.2 Interrupt Handling at the CPU ... 1512
8.2.3 Software Interrupt Handling Options ... 1513
8.3 Interrupt Handling Inside VIM ... 1514
8.3.1 VIM Interrupt Channel Mapping ... 1515
8.3.2 VIM Input Channel Management ... 1517
8.4 Interrupt Vector Table (VIM RAM) ... 1518
8.4.1 Interrupt Vector Table Operation ... 1518
8.4.2 VIM ECC Syndrome ... 1519
8.4.3 Interrupt Vector Table Initialization ... 1520
8.4.4 Interrupt Vector Table ECC Testing ... 1520
8.5 VIM Wakeup Interrupt ... 1523
8.6 Capture Event Sources ... 1524
8.7 Examples ... 1524
8.7.1 Examples - Configure CPU To Receive Interrupts ... 1524
8.7.2 Examples - Register Vector Interrupt and Index Interrupt Handling ... 1525
8.8 VIM Control Registers ... 1527
8.8.1 Interrupt Vector Table ECC Status Register (ECCSTAT) ... 1528
8.8.2 Interrupt Vector Table ECC Control Register (ECCCTL) ... 1529
8.8.3 Uncorrectable Error Address Register (UERRADDR) ... 1530
8.8.4 Fallback Vector Address Register (FBVECADDR) ... 1530
8.8.5 Single Bit Error Address Register (SBERRADDR) ... 1531

8.8.6 VIM Offset Vector Registers 1531
8.8.7 IRQ Index Offset Vector Register (IRQINDEX) 1532
8.8.8 FIQ Index Offset Vector Registers (FIQINDEX) 1532
8.8.9 FIQ/IRQ Program Control Registers (FIRQPR[0:3]) 1533
8.8.10 Pending Interrupt Read Location Registers (INTREQ[0:3]) 1534
8.8.11 Interrupt Enable Set Registers (REQENASET[0:3]) 1535
8.8.12 Interrupt Enable Clear Registers (REQENACLR[0:3]) 1536
8.8.13 Wake-Up Enable Set Registers (WAKEENASET[0:3]) 1537
8.8.14 Wake-Up Enable Clear Registers (WAKEENACLR[0:3]) 1538
8.8.15 IRQ Interrupt Vector Register (IRQVECREG) 1539
8.8.16 FIQ Interrupt Vector Register (FIQVECREG) 1539
8.8.17 Capture Event Register (CAPEVT) 1540
8.8.18 VIM Interrupt Control Registers (CHANCTRL[0:31]) 1541
8.9 Interrupt Request Assignments 1542
9 DSP Subsystem C674x 1543
9.1 Introduction 1544
9.2 TMS320C674x Megamodule 1545
9.2.1 Internal Memory Controllers 1545
9.2.2 Internal Peripherals 1545
9.3 Memory Map 1546
9.4 Advanced Event Triggering (AET) 1546
9.5 DSP Event Assignment 1547
10 DSS_L3 Memory Organization 1548
10.1 DSS_L3 Memory Organization for 14xx 1549
10.1.1 Functional Description 1549
10.1.2 Memory Auto Initialization 1550
10.1.3 Memory TAB Register 1551
10.2 DSS_L3 Memory Organization for 16xx/18xx 1551
10.2.1 Functional Description 1551
10.2.2 Memory Auto-Initialization 1552
10.2.3 Memory TAB Register 1552
10.3 DSS_L3 Memory Organization for 68xx 1552
10.3.1 Functional Description 1553
10.3.2 Memory Auto-Initialization 1553
10.3.3 Memory TAB Register 1554
11 Handshake RAM (HSRAM) 1555
12 Enhanced Direct Memory Access (EDMA) 1556
12.1 EDMA Module Overview 1557
12.1.1 EDMA Features 1558
12.2 EDMA Controller Functional Description 1559
12.2.1 Block Diagram 1559
12.2.2 Types of EDMA controller Transfers 1562
12.2.3 Parameter RAM (PaRAM) 1564
12.2.4 Initiating a DMA Transfer 1576
12.2.5 Completion of a DMA Transfer 1579
12.2.6 Event, Channel, and PaRAM Mapping 1580
12.2.7 EDMA Channel Controller Regions 1582
12.2.8 Chaining EDMA Channels 1584
12.2.9 EDMA Interrupts 1586
12.2.10 Memory Protection 1594
12.2.11 Event Queue(s) 1599
12.2.12 EDMA Transfer Controller (EDMA_TPTC) 1601

12.2.13 Event Dataflow 1603
12.2.14 EDMA Controller Prioritization 1604
12.2.15 Emulation Considerations 1606
12.3 EDMA Transfer Examples 1607
12.3.1 Block Move Example 1607
12.3.2 Subframe Extraction Example 1609
12.3.3 Data Sorting Example 1610
12.3.4 Setting Up an EDMA Transfer 1612
12.4 EDMA Debug Checklist and Programming Tips 1614
12.4.1 EDMA Debug Checklist 1614
12.4.2 EDMA Programming Tips 1615
12.5 EDMA Request Map 1615
12.6 EDMA Register Manual 1616
12.6.1 EDMA Registers 1616
13 ADC Buffer 1845
13.1 Functional Description 1846
13.1.1 DFE Data Write Operation 1846
13.1.2 Support for Hardware in Loop (HIL) 1848
13.1.3 Test Pattern Generator Support 1849
13.1.4 ADC Buffer Data Formats 1849
14 High-Speed Interface (HSI) 1851
14.1 Overview 1852
14.2 CBUFF 1852
14.2.1 CBUFF Overview 1852
14.2.2 CBUFF Sequencing 1853
14.2.3 Sequence-Related CBUFF Configuration Fields 1855
14.2.4 CBUFF Linklist Concept 1855
14.2.5 CBUFF Interrupts 1858
14.3 LVDS 1859
14.3.1 LVDS Overview 1859
14.3.2 LVDS Programming Sequence 1859
14.3.3 CBUFF and LVDS Registers 1861
14.4 CSI2 1998
14.4.1 CSI2 Overview 1998
14.4.2 CSI2 Protocol Engine and Phy 2009
14.4.3 CSI2 Programming Sequence 2021
14.4.4 CSI2_PROTOCOL_ENGINE Registers 2025
14.4.5 CSI2_PHY Registers 2095
15 Hardware Accelerator 2113
15.1 Introduction 2114
15.2 Register Map 2114
16 Real Time Interrupt (RTI) and RTI With Digital Watchdog Timer (WDT) 2115
16.1 Overview 2116
16.1.1 Features 2116
16.1.2 Industry Standard Compliance Statement 2116
16.2 Module Operation 2117
16.2.1 Counter Operation 2117
16.2.2 Interrupt/DMA Requests 2119
16.2.3 Digital Watchdog (DWD) 2120
16.2.4 Halting Debug Mode Behaviour 2123
16.3 RTI Control Registers 2124
16.3.1 RTI Global Control Register (RTIGCTRL) 2125

16.3.2 RTI Timebase Control Register (RTITBCTRL) 2126
16.3.3 RTI Capture Control Register (RTICAPCTRL) 2127
16.3.4 RTI Compare Control Register (RTICOMPCTRL) 2128
16.3.5 RTI Free Running Counter 0 Register (RTIFRC0) 2129
16.3.6 RTI Up Counter 0 Register (RTIUC0) 2129
16.3.7 RTI Compare Up Counter 0 Register (RTICPUC0) 2130
16.3.8 RTI Capture Free Running Counter 0 Register (RTICAFRC0) 2130
16.3.9 RTI Capture Up Counter 0 Register (RTICAUC0) 2131
16.3.10 RTI Free Running Counter 1 Register (RTIFRC1) 2131
16.3.11 RTI Up Counter 1 Register (RTIUC1) 2132
16.3.12 RTI Compare Up Counter 1 Register (RTICPUC1) 2133
16.3.13 RTI Capture Free Running Counter 1 Register (RTICAFRC1) 2134
16.3.14 RTI Capture Up Counter 1 Register (RTICAUC1) 2134
16.3.15 RTI Compare 0 Register (RTICOMP0) 2135
16.3.16 RTI Update Compare 0 Register (RTIUDCP0) 2135
16.3.17 RTI Compare 1 Register (RTICOMP1) 2136
16.3.18 RTI Update Compare 1 Register (RTIUDCP1) 2136
16.3.19 RTI Compare 2 Register (RTICOMP2) 2137
16.3.20 RTI Update Compare 2 Register (RTIUDCP2) 2137
16.3.21 RTI Compare 3 Register (RTICOMP3) 2138
16.3.22 RTI Update Compare 3 Register (RTIUDCP3) 2138
16.3.23 RTI Timebase Low Compare Register (RTITBLCOMP) 2139
16.3.24 RTI Timebase High Compare Register (RTITBHCOMP) 2139
16.3.25 RTI Set Interrupt Enable Register (RTISETINTENA) 2140
16.3.26 RTI Clear Interrupt Enable Register (RTICLEARINTENA) 2142
16.3.27 RTI Interrupt Flag Register (RTIINTFLAG) 2144
16.3.28 Digital Watchdog Control Register (RTIDWDCTRL) 2145
16.3.29 Digital Watchdog Preload Register (RTIDWDPRLD) 2146
16.3.30 Watchdog Status Register (RTIWDSTATUS) 2147
16.3.31 RTI Watchdog Key Register (RTIWDKEY) 2148
16.3.32 RTI Digital Watchdog Down Counter (RTIDWDCNTR) 2149
16.3.33 Digital Windowed Watchdog Reaction Control (RTIWWDRXNCTRL) 2149
16.3.34 Digital Windowed Watchdog Window Size Control (RTIWWDSIZECTRL) 2150
16.3.35 RTI Compare Interrupt Clear Enable Register (RTIINTCLRENABLE) 2151
16.3.36 RTI Compare 0 Clear Register (RTICMP0CLR) 2152
16.3.37 RTI Compare 1 Clear Register (RTICMP1CLR) 2152
16.3.38 RTI Compare 2 Clear Register (RTICMP2CLR) 2153
16.3.39 RTI Compare 3 Clear Register (RTICMP3CLR) 2153

17 General Purpose I/0 (GIO) 2154

17.1 Overview 2155
17.2 Quick Start Guide 2155
17.3 Functional Description of GIO Module 2157
17.3.1 I/O Functions 2157
17.3.2 Interrupt Function 2158
17.3.3 GIO Block Diagram 2158
17.4 Device Modes of Operation 2160
17.4.1 Emulation Mode 2160
17.4.2 Power-Down Mode (Low-Power Mode) 2160
17.5 MSS_GIO Registers 2161
17.5.1 GIOGCR Register (Offset = 0h) [reset = 0h] 2163
17.5.2 GIOPWDN Register (Offset = 4h) [reset = 0h] 2164
17.5.3 GIOINTDET Register (Offset = 8h) [reset = 0h] 2165
17.5.4 GIOPOL Register (Offset = Ch) [reset = 0h] 2166

17.5.5 GIOENASET Register (Offset = 10h) [reset = 0h] 2167
17.5.6 GIOENACLR Register (Offset = 14h) [reset = 0h] 2168
17.5.7 GIOLVLSET Register (Offset = 18h) [reset = 0h] 2169
17.5.8 GIOLVLCLR Register (Offset = 1Ch) [reset = 0h] 2170
17.5.9 GIOFLG Register (Offset = 20h) [reset = 0h] 2171
17.5.10 GIOOFFA Register (Offset = 24h) [reset = 0h] 2172
17.5.11 GIOOFFB Register (Offset = 28h) [reset = 0h] 2173
17.5.12 GIOEMUA Register (Offset = 2Ch) [reset = 0h] 2174
17.5.13 GIOEMUB Register (Offset = 30h) [reset = 0h] 2175
17.5.14 GIODIRA Register (Offset = 34h) [reset = 0h] 2176
17.5.15 GIODINA Register (Offset = 38h) [reset = 0h] 2177
17.5.16 GIODOUTA Register (Offset = 3Ch) [reset = 0h] 2178
17.5.17 GIOSETA Register (Offset = 40h) [reset = 0h] 2179
17.5.18 GIOCLRA Register (Offset = 44h) [reset = 0h] 2180
17.5.19 GIOPDRA Register (Offset = 48h) [reset = 0h] 2181
17.5.20 GIOPULDISA Register (Offset = 4Ch) [reset = 0h] 2182
17.5.21 GIOPSLA Register (Offset = 50h) [reset = 0h] 2183
17.5.22 GIODIRB Register (Offset = 54h) [reset = 0h] 2184
17.5.23 GIODINB Register (Offset = 58h) [reset = 0h] 2185
17.5.24 GIODOUTB Register (Offset = 5Ch) [reset = 0h] 2186
17.5.25 GIOSETB Register (Offset = 60h) [reset = 0h] 2187
17.5.26 GIOCLRB Register (Offset = 64h) [reset = 0h] 2188
17.5.27 GIOPDRB Register (Offset = 68h) [reset = 0h] 2189
17.5.28 GIOPULDISB Register (Offset = 6Ch) [reset = 0h] 2190
17.5.29 GIOPSLB Register (Offset = 70h) [reset = 0h] 2191
17.5.30 GIODIRC Register (Offset = 74h) [reset = 0h] 2192
17.5.31 GIODINC Register (Offset = 78h) [reset = 0h] 2193
17.5.32 GIODOUTC Register (Offset = 7Ch) [reset = 0h] 2194
17.5.33 GIOSETC Register (Offset = 80h) [reset = 0h] 2195
17.5.34 GIOCLRC Register (Offset = 84h) [reset = 0h] 2196
17.5.35 GIOPDRC Register (Offset = 88h) [reset = 0h] 2197
17.5.36 GIOPULDISC Register (Offset = 8Ch) [reset = 0h] 2198
17.5.37 GIOPSLC Register (Offset = 90h) [reset = 0h] 2199
17.5.38 GIODIRD Register (Offset = 94h) [reset = 0h] 2200
17.5.39 GIODIND Register (Offset = 98h) [reset = 0h] 2201
17.5.40 GIODOUTD Register (Offset = 9Ch) [reset = 0h] 2202
17.5.41 GIOSETD Register (Offset = A0h) [reset = 0h] 2203
17.5.42 GIOCLRD Register (Offset = A4h) [reset = 0h] 2204
17.5.43 GIOPDRD Register (Offset = A8h) [reset = 0h] 2205
17.5.44 GIOPULDISD Register (Offset = ACh) [reset = 0h] 2206
17.5.45 GIOPSLD Register (Offset = B0h) [reset = 0h] 2207
17.5.46 GIODIRE Register (Offset = B4h) [reset = 0h] 2208
17.5.47 GIODINE Register (Offset = B8h) [reset = 0h] 2209
17.5.48 GIODOUTE Register (Offset = BCh) [reset = 0h] 2210
17.5.49 GIOSETE Register (Offset = C0h) [reset = 0h] 2211
17.5.50 GIOCLRE Register (Offset = C4h) [reset = 0h] 2212
17.5.51 GIOPDRE Register (Offset = C8h) [reset = 0h] 2213
17.5.52 GIOPULDISE Register (Offset = CCh) [reset = 0h] 2214
17.5.53 GIOPSLE Register (Offset = D0h) [reset = 0h] 2215
17.5.54 GIODIRF Register (Offset = D4h) [reset = 0h] 2216
17.5.55 GIODINF Register (Offset = D8h) [reset = 0h] 2217
17.5.56 GIODOUTF Register (Offset = DCh) [reset = 0h] 2218
17.5.57 GIOSETF Register (Offset = E0h) [reset = 0h] 2219

17.5.58 GIOCLRF Register (Offset = E4h) [reset = 0h] 2220
17.5.59 GIOPDRF Register (Offset = E8h) [reset = 0h] 2221
17.5.60 GIOPULDISF Register (Offset = ECh) [reset = 0h] 2222
17.5.61 GIOPSLF Register (Offset = F0h) [reset = 0h] 2223
17.5.62 GIODIRG Register (Offset = F4h) [reset = 0h] 2224
17.5.63 GIODING Register (Offset = F8h) [reset = 0h] 2225
17.5.64 GIODOUTG Register (Offset = FCh) [reset = 0h] 2226
17.5.65 GIOSETG Register (Offset = 100h) [reset = 0h] 2227
17.5.66 GIOCLRG Register (Offset = 104h) [reset = 0h] 2228
17.5.67 GIOPDRG Register (Offset = 108h) [reset = 0h] 2229
17.5.68 GIOPULDISG Register (Offset = 10Ch) [reset = 0h] 2230
17.5.69 GIOPSLG Register (Offset = 110h) [reset = 0h] 2231
17.5.70 GIODIRH Register (Offset = 114h) [reset = 0h] 2232
17.5.71 GIODINH Register (Offset = 118h) [reset = 0h] 2233
17.5.72 GIODOUTH Register (Offset = 11Ch) [reset = 0h] 2234
17.5.73 GIOSETH Register (Offset = 120h) [reset = 0h] 2235
17.5.74 GIOCLRH Register (Offset = 124h) [reset = 0h] 2236
17.5.75 GIOPDRH Register (Offset = 128h) [reset = 0h] 2237
17.5.76 GIOPULDISH Register (Offset = 12Ch) [reset = 0h] 2238
17.5.77 GIOPSLH Register (Offset = 130h) [reset = 0h] 2239
17.5.78 GIOSRCA Register (Offset = 134h) [reset = 0h] 2240
17.5.79 GIOSRCB Register (Offset = 138h) [reset = 0h] 2241
17.5.80 GIOSRCC Register (Offset = 13Ch) [reset = 0h] 2242
17.5.81 GIOSRCD Register (Offset = 140h) [reset = 0h] 2243
17.5.82 GIOSRCE Register (Offset = 144h) [reset = 0h] 2244
17.5.83 GIOSRCF Register (Offset = 148h) [reset = 0h] 2245
17.5.84 GIOSRCG Register (Offset = 14Ch) [reset = 0h] 2246
17.5.85 GIOSRCH Register (Offset = 150h) [reset = 0h] 2247
17.6 I/O Control Summary 2248
18 Mailbox 2249
18.1 Maibox Message Scheme 2250
18.2 Mailbox Registers 2251
18.2.1 INT_MASK Register (Offset = 0h) [reset = 0h] 2252
18.2.2 INT_MASK_SET Register (Offset = 8h) [reset = 0h] 2253
18.2.3 INT_MASK_CLR Register (Offset = 10h) [reset = 0h] 2254
18.2.4 INT_STS_CLR Register (Offset = 18h) [reset = 0h] 2255
18.2.5 INT_ACK Register (Offset = 20h) [reset = 0h] 2256
18.2.6 INT_TRIG Register (Offset = 28h) [reset = 0h] 2257
18.2.7 INT_STS_MASKED Register (Offset = 30h) [reset = 0h] 2258
18.2.8 INT_STS_RAW Register (Offset = 38h) [reset = 0h] 2259
19 Data Modification Module (DMM) 2260
19.1 Overview 2261
19.1.1 Features 2261
19.1.2 Block Diagram 2261
19.2 Module Operation 2262
19.2.1 Data Format 2262
19.2.2 Data Port 2264
19.2.3 Error Handling 2265
19.2.4 Interrupts 2266
19.3 Control Registers 2267
19.3.1 DMM Global Control Register (DMMGLBCTRL) 2268
19.3.2 DMM Interrupt Set Register (DMMINTSET) 2270
19.3.3 DMM Interrupt Clear Register (DMMINTCLR) 2274

19.3.4 DMM Interrupt Level Register (DMMINTLVL) 2279
19.3.5 DMM Interrupt Flag Register (DMMINTFLG) 2281
19.3.6 DMM Interrupt Offset 1 Register (DMMOFF1) 2285
19.3.7 DMM Interrupt Offset 2 Register (DMMOFF2) 2286
19.3.8 DMM Direct Data Mode Destination Register (DMMDDMDEST) 2287
19.3.9 DMM Direct Data Mode Blocksize Register (DMMDDMBL) 2287
19.3.10 DMM Direct Data Mode Pointer Register (DMMDDMPT) 2288
19.3.11 DMM Direct Data Mode Interrupt Pointer Register (DMMINTPT) 2288
19.3.12 DMM Destination x Region 1 (DMMDESTxREG1) 2289
19.3.13 DMM Destination x Blocksize 1 (DMMDESTxBL1) 2290
19.3.14 DMM Destination x Region 2 (DMMDESTxREG2) 2291
19.3.15 DMM Destination x Blocksize 2 (DMMDESTxBL2) 2292
19.3.16 DMM Pin Control 0 (DMMPC0) 2293
19.3.17 DMM Pin Control 1 (DMMPC1) 2294
19.3.18 DMM Pin Control 2 (DMMPC2) 2296
19.3.19 DMM Pin Control 3 (DMMPC3) 2297
19.3.20 DMM Pin Control 4 (DMMPC4) 2298
19.3.21 DMM Pin Control 5 (DMMPC5) 2300
19.3.22 DMM Pin Control 6 (DMMPC6) 2301
19.3.23 DMM Pin Control 7 (DMMPC7) 2303
19.3.24 DMM Pin Control 8 (DMMPC8) 2304

20 Enhanced Pulse Width Modulator (ePWM) 2306

20.1 Introduction 2307
20.1.1 Submodule Overview 2307
20.1.2 Register Mapping 2310
20.2 ePWM Submodules 2311
20.2.1 Overview 2311
20.2.2 Time-Base (TB) Submodule 2313
20.2.3 Counter-Compare (CC) Submodule 2321
20.2.4 Action-Qualifier (AQ) Submodule 2326
20.2.5 Dead-Band Generator (DB) Submodule 2339
20.2.6 PWM-Chopper (PC) Submodule 2344
20.2.7 Trip-Zone (TZ) Submodule 2348
20.2.8 Event-Trigger (ET) Submodule 2354
20.2.9 Digital Compare (DC) Submodule 2359
20.2.10 Proper Interrupt Initialization Procedure 2365
20.3 Application Examples 2366
20.3.1 Overview of Multiple Modules 2366
20.3.2 Key Configuration Capabilities 2366
20.3.3 Controlling Multiple Buck Converters With Independent Frequencies 2368
20.3.4 Controlling Multiple Buck Converters With Same Frequencies 2371
20.3.5 Controlling Multiple Half H-Bridge (HHB) Converters 2374
20.3.6 Controlling Dual 3-Phase Inverters for Motors (ACI and PMSM) 2376
20.3.7 Practical Applications Using Phase Control Between PWM Modules 2379
20.4 MSS_ETPWM1 Registers 2381
20.4.1 TBCTL_TBSTS Register (Offset = 0h) [reset = 0h] 2383
20.4.2 TBPHS Register (Offset = 4h) [reset = 0h] 2385
20.4.3 TBCTR_TBPRD Register (Offset = 8h) [reset = 0h] 2386
20.4.4 CMPCTL Register (Offset = Ch) [reset = 0h] 2387
20.4.5 CMPA Register (Offset = 10h) [reset = 0h] 2389
20.4.6 CMPB_AQCTLA Register (Offset = 14h) [reset = 0h] 2390
20.4.7 AQCTLB_AQSFRC Register (Offset = 18h) [reset = 0h] 2392
20.4.8 AQCSFRC_DBCTL Register (Offset = 1Ch) [reset = 0h] 2394

20.4.9 DBRED_DBFED Register (Offset = 20h) [reset = 0h] 2396
20.4.10 TZSEL_TZDCSEL Register (Offset = 24h) [reset = 0h] 2397
20.4.11 TZCTL_TZEINT Register (Offset = 28h) [reset = 0h] 2399
20.4.12 TZFLG_TZCLR Register (Offset = 2Ch) [reset = 0h] 2401
20.4.13 TZFRC_ETSEL Register (Offset = 30h) [reset = 0h] 2403
20.4.14 ETPS_ETFLG Register (Offset = 34h) [reset = 0h] 2405
20.4.15 ETCLR_ETFRC Register (Offset = 38h) [reset = 0h] 2407
20.4.16 PCCTL Register (Offset = 3Ch) [reset = 0h] 2408
20.4.17 Reserved1 Register (Offset = 40h) [reset = 0h] 2409
20.4.18 Reserved2 Register (Offset = 44h) [reset = 0h] 2410
20.4.19 Reserved3 Register (Offset = 48h) [reset = 0h] 2411
20.4.20 Reserved4 Register (Offset = 4Ch) [reset = 0h] 2412
20.4.21 Reserved5 Register (Offset = 50h) [reset = 0h] 2413
20.4.22 Reserved6 Register (Offset = 54h) [reset = 0h] 2414
20.4.23 Reserved7 Register (Offset = 58h) [reset = 0h] 2415
20.4.24 Reserved8 Register (Offset = 5Ch) [reset = 0h] 2416
20.4.25 DCTRIPSEL_DCACTL Register (Offset = 60h) [reset = 0h] 2417
20.4.26 DCBCTL_DCFCTL Register (Offset = 64h) [reset = 0h] 2419
20.4.27 DCCAPCTL_DCFOFFSET Register (Offset = 68h) [reset = 0h] 2420
20.4.28 DCFOFFSETCNT_DCFWINDOW Register (Offset = 6Ch) [reset = 0h] 2421
20.4.29 DCFWINDOWCNT_DCCAP Register (Offset = 70h) [reset = 0h] 2422

21 Controller Area Network (DCAN) 2423

21.1 Overview 2424
21.1.1 Features 2424
21.1.2 Functional Description 2424
21.2 CAN Blocks 2425
21.2.1 CAN Core 2425
21.2.2 Message RAM 2425
21.2.3 Message Handler 2425
21.2.4 Message RAM Interface 2426
21.2.5 Register and Message Object Access 2426
21.2.6 Dual Clock Source 2426
21.3 CAN Bit Timing 2427
21.3.1 Bit Time and Bit Rate 2427
21.3.2 DCAN Bit Timing Registers 2429
21.4 CAN Module Configuration 2431
21.4.1 DCAN RAM Initialization Through Hardware 2431
21.4.2 CAN Module Initialization 2431
21.5 Message RAM 2434
21.5.1 Structure of Message Objects 2434
21.5.2 Addressing Message Objects in RAM 2436
21.5.3 Message RAM Representation in Debug/Suspend Mode 2437
21.5.4 Message RAM Representation in Direct Access Mode 2437
21.5.5 ECC RAM 2438
21.6 Message Interface Register Sets 2439
21.6.1 Message Interface Register Sets 1 and 2 2439
21.6.2 Using Message Interface Register Sets 1 and 2 2440
21.6.3 Message Interface Register 3 2441
21.7 Message Object Configurations 2442
21.7.1 Configuration of a Transmit Object for Data Frames 2442
21.7.2 Configuration of a Transmit Object for Remote Frames 2442
21.7.3 Configuration of a Single Receive Object for Data Frames 2442
21.7.4 Configuration of a Single Receive Object for Remote Frames 2443

21.7.5 Configuration of a FIFO Buffer 2443
21.7.6 Reconfiguration of Message Objects for the Reception of Frames 2443
21.7.7 Reconfiguration of Message Objects for the Reception of Frames 2443
21.8 Message Handling 2444
21.8.1 Message Handler Overview 2444
21.8.2 Receive/Transmit Priority 2444
21.8.3 Transmission of Messages in Event Driven CAN Communication 2445
21.8.4 Updating a Transmit Object 2445
21.8.5 Changing a Transmit Object 2445
21.8.6 Acceptance Filtering of Received Messages 2446
21.8.7 Reception of Data Frames 2446
21.8.8 Reception of Remote Frames 2446
21.8.9 Reading Received Messages 2446
21.8.10 Requesting New Data for a Receive Object 2447
21.8.11 Storing Received Messages in FIFO Buffers 2447
21.8.12 Reading from a FIFO Buffer 2447
21.9 CAN Message Transfer 2449
21.9.1 Automatic Retransmission 2449
21.9.2 Auto-Bus-On 2450
21.10 Interrupt Functionality 2450
21.10.1 Message Object Interrupts 2450
21.10.2 Status Change Interrupts 2451
21.10.3 Error Interrupts 2451
21.11 Global Power Down Mode 2452
21.11.1 Entering Global Power Down Mode 2452
21.11.2 Wakeup From Global Power Down Mode 2452
21.12 Local Power Down Mode 2453
21.12.1 Entering Local Power Down Mode 2453
21.12.2 Wakeup From Local Power Down 2453
21.13 GIO Support 2453
21.14 Test Modes 2455
21.14.1 Silent Mode 2455
21.14.2 Loop Back Mode 2456
21.14.3 External Loop Back Mode 2457
21.14.4 Loop Back Combined with Silent Mode 2458
21.14.5 Software Control of CAN_TX Pin 2458
21.15 SECDED Mechanism 2459
21.15.1 Behavior on Single Bit Error 2459
21.15.2 Behavior on Double Bit Error 2459
21.15.3 SECDED Testing 2459
21.16 Debug/Suspend Mode 2460
21.17 MSS_DCAN Registers 2461
21.17.1 CTL Register (Offset = 0h) [reset = 1401h] 2464
21.17.2 ES Register (Offset = 4h) [reset = 7h] 2467
21.17.3 ERRC Register (Offset = 8h) [reset = 0h] 2469
21.17.4 BTR Register (Offset = Ch) [reset = 2301h] 2470
21.17.5 INT Register (Offset = 10h) [reset = 0h] 2471
21.17.6 TEST Register (Offset = 14h) [reset = 80h] 2472
21.17.7 Reserved_1 Register (Offset = 18h) [reset = 0h] 2473
21.17.8 PERR Register (Offset = 1Ch) [reset = 0h] 2474
21.17.9 DCAN_REV_ID Register (Offset = 20h) [reset = 0h] 2475
21.17.10 ECCDIAG Register (Offset = 24h) [reset = 0h] 2476
21.17.11 ECCDIAG_STAT Register (Offset = 28h) [reset = 0h] 2477

21.17.12 ECC_CS Register (Offset = 2Ch) [reset = 0h] 2478
21.17.13 ECC_SERR Register (Offset = 30h) [reset = 0h] 2479
21.17.14 TTCAN1 Register (Offset = 34h) [reset = 0h] 2480
21.17.15 TTCAN2 Register (Offset = 38h) [reset = 0h] 2481
21.17.16 TTCAN3 Register (Offset = 3Ch) [reset = 0h] 2482
21.17.17 TTCAN4 Register (Offset = 40h) [reset = 0h] 2483
21.17.18 TTCAN5 Register (Offset = 44h) [reset = 0h] 2484
21.17.19 TTCAN6 Register (Offset = 48h) [reset = 0h] 2485
21.17.20 TTCAN7 Register (Offset = 4Ch) [reset = 0h] 2486
21.17.21 TTCAN8 Register (Offset = 50h) [reset = 0h] 2487
21.17.22 TTCAN9 Register (Offset = 54h) [reset = 0h] 2488
21.17.23 TTCAN10 Register (Offset = 58h) [reset = 0h] 2489
21.17.24 TTCAN11 Register (Offset = 5Ch) [reset = 0h] 2490
21.17.25 TTCAN12 Register (Offset = 60h) [reset = 0h] 2491
21.17.26 TTCAN13 Register (Offset = 64h) [reset = 0h] 2492
21.17.27 TTCAN14 Register (Offset = 68h) [reset = 0h] 2493
21.17.28 TTCAN15 Register (Offset = 6Ch) [reset = 0h] 2494
21.17.29 TTCAN16 Register (Offset = 70h) [reset = 0h] 2495
21.17.30 TTCAN17 Register (Offset = 74h) [reset = 0h] 2496
21.17.31 TTCAN18 Register (Offset = 78h) [reset = 0h] 2497
21.17.32 TTCAN19 Register (Offset = 7Ch) [reset = 0h] 2498
21.17.33 ABOTR Register (Offset = 80h) [reset = 0h] 2499
21.17.34 TXRQ_X Register (Offset = 84h) [reset = 0h] 2500
21.17.35 TXRQ12 Register (Offset = 88h) [reset = 0h] 2501
21.17.36 TXRQ34 Register (Offset = 8Ch) [reset = 0h] 2502
21.17.37 TXRQ56 Register (Offset = 90h) [reset = 0h] 2503
21.17.38 TXRQ78 Register (Offset = 94h) [reset = 0h] 2504
21.17.39 NWDAT_X Register (Offset = 98h) [reset = 0h] 2505
21.17.40 NWDAT12 Register (Offset = 9Ch) [reset = 0h] 2506
21.17.41 NWDAT34 Register (Offset = A0h) [reset = 0h] 2507
21.17.42 NWDAT56 Register (Offset = A4h) [reset = 0h] 2508
21.17.43 NWDAT78 Register (Offset = A8h) [reset = 0h] 2509
21.17.44 INTPND_X Register (Offset = ACh) [reset = 0h] 2510
21.17.45 INTPND12 Register (Offset = B0h) [reset = 0h] 2511
21.17.46 INTPND34 Register (Offset = B4h) [reset = 0h] 2512
21.17.47 INTPND56 Register (Offset = B8h) [reset = 0h] 2513
21.17.48 INTPND78 Register (Offset = BCh) [reset = 0h] 2514
21.17.49 MSGVAL_X Register (Offset = C0h) [reset = 0h] 2515
21.17.50 MSGVAL12 Register (Offset = C4h) [reset = 0h] 2516
21.17.51 MSGVAL34 Register (Offset = C8h) [reset = 0h] 2517
21.17.52 MSGVAL56 Register (Offset = CCh) [reset = 0h] 2518
21.17.53 MSGVAL78 Register (Offset = D0h) [reset = 0h] 2519
21.17.54 Reserved_2 Register (Offset = D4h) [reset = 0h] 2520
21.17.55 INTMUX12 Register (Offset = D8h) [reset = 0h] 2521
21.17.56 INTMUX34 Register (Offset = DCh) [reset = 0h] 2522
21.17.57 INTMUX56 Register (Offset = E0h) [reset = 0h] 2523
21.17.58 INTMUX78 Register (Offset = E4h) [reset = 0h] 2524
21.17.59 Reserved_3 Register (Offset = E8h) [reset = 0h] 2525
21.17.60 Reserved_4 Register (Offset = ECh) [reset = 0h] 2526
21.17.61 Reserved_5 Register (Offset = F0h) [reset = 0h] 2527
21.17.62 Reserved_6 Register (Offset = F4h) [reset = 0h] 2528
21.17.63 Reserved_7 Register (Offset = F8h) [reset = 0h] 2529
21.17.64 Reserved_8 Register (Offset = FCh) [reset = 0h] 2530

21.17.65 IF1CMD Register (Offset = 100h) [reset = 1h] 2531
21.17.66 IF1MSK Register (Offset = 104h) [reset = FFFFFFFFh] 2534
21.17.67 IF1ARB Register (Offset = 108h) [reset = 0h] 2535
21.17.68 IF1MCTL Register (Offset = 10Ch) [reset = 0h] 2536
21.17.69 IF1DATA Register (Offset = 110h) [reset = 0h] 2538
21.17.70 IF1DATB Register (Offset = 114h) [reset = 0h] 2539
21.17.71 Reserved_9 Register (Offset = 118h) [reset = 0h] 2540
21.17.72 Reserved_10 Register (Offset = 11Ch) [reset = 0h] 2541
21.17.73 IF2CMD Register (Offset = 120h) [reset = 1h] 2542
21.17.74 IF2MSK Register (Offset = 124h) [reset = FFFFFFFFh] 2545
21.17.75 IF2ARB Register (Offset = 128h) [reset = 0h] 2546
21.17.76 IF2MCTL Register (Offset = 12Ch) [reset = 0h] 2547
21.17.77 IF2DATA Register (Offset = 130h) [reset = 0h] 2549
21.17.78 IF2DATB Register (Offset = 134h) [reset = 0h] 2550
21.17.79 Reserved_11 Register (Offset = 138h) [reset = 0h] 2551
21.17.80 Reserved_12 Register (Offset = 13Ch) [reset = 0h] 2552
21.17.81 IF3OBS Register (Offset = 140h) [reset = 0h] 2553
21.17.82 IF3MSK Register (Offset = 144h) [reset = FFFFFFFFh] 2555
21.17.83 IF3ARB Register (Offset = 148h) [reset = 0h] 2556
21.17.84 IF3MCTL Register (Offset = 14Ch) [reset = 0h] 2557
21.17.85 IF3DATA Register (Offset = 150h) [reset = 0h] 2559
21.17.86 IF3DATB Register (Offset = 154h) [reset = 0h] 2560
21.17.87 Reserved_13 Register (Offset = 158h) [reset = 0h] 2561
21.17.88 Reserved_14 Register (Offset = 15Ch) [reset = 0h] 2562
21.17.89 IF3UPD12 Register (Offset = 160h) [reset = 0h] 2563
21.17.90 IF3UPD34 Register (Offset = 164h) [reset = 0h] 2564
21.17.91 IF3UPD56 Register (Offset = 168h) [reset = 0h] 2565
21.17.92 IF3UPD78 Register (Offset = 16Ch) [reset = 0h] 2566
21.17.93 Reserved_15 Register (Offset = 170h) [reset = 0h] 2567
21.17.94 Reserved_16 Register (Offset = 174h) [reset = 0h] 2568
21.17.95 Reserved_17 Register (Offset = 178h) [reset = 0h] 2569
21.17.96 Reserved_18 Register (Offset = 17Ch) [reset = 0h] 2570
21.17.97 Reserved_19 Register (Offset = 180h) [reset = 0h] 2571
21.17.98 Reserved_20 Register (Offset = 184h) [reset = 0h] 2572
21.17.99 Reserved_21 Register (Offset = 188h) [reset = 0h] 2573
21.17.100 Reserved_22 Register (Offset = 18Ch) [reset = 0h] 2574
21.17.101 Reserved_23 Register (Offset = 190h) [reset = 0h] 2575
21.17.102 Reserved_24 Register (Offset = 194h) [reset = 0h] 2576
21.17.103 Reserved_25 Register (Offset = 198h) [reset = 0h] 2577
21.17.104 Reserved_26 Register (Offset = 19Ch) [reset = 0h] 2578
21.17.105 Reserved_27 Register (Offset = 1A0h) [reset = 0h] 2579
21.17.106 Reserved_28 Register (Offset = 1A4h) [reset = 0h] 2580
21.17.107 Reserved_29 Register (Offset = 1A8h) [reset = 0h] 2581
21.17.108 Reserved_30 Register (Offset = 1ACh) [reset = 0h] 2582
21.17.109 Reserved_31 Register (Offset = 1B0h) [reset = 0h] 2583
21.17.110 Reserved_32 Register (Offset = 1B4h) [reset = 0h] 2584
21.17.111 Reserved_33 Register (Offset = 1B8h) [reset = 0h] 2585
21.17.112 Reserved_34 Register (Offset = 1BCh) [reset = 0h] 2586
21.17.113 Reserved_35 Register (Offset = 1C0h) [reset = 0h] 2587
21.17.114 Reserved_36 Register (Offset = 1C4h) [reset = 0h] 2588
21.17.115 Reserved_37 Register (Offset = 1C8h) [reset = 0h] 2589
21.17.116 Reserved_38 Register (Offset = 1CCh) [reset = 0h] 2590
21.17.117 Reserved_39 Register (Offset = 1D0h) [reset = 0h] 2591

21.17.118 Reserved_40 Register (Offset = 1D4h) [reset = 0h] ... 2592
21.17.119 Reserved_41 Register (Offset = 1D8h) [reset = 0h] ... 2593
21.17.120 Reserved_42 Register (Offset = 1DCh) [reset = 0h] ... 2594
21.17.121 TIOC Register (Offset = 1E0h) [reset = 0h] ... 2595
21.17.122 RIOC Register (Offset = 1E4h) [reset = 0h] ... 2597

22 Modular Controller Area Network (MCAN) ... 2599
22.1 MCAN (14xx,16xx,18xx) ... 2600
22.1.1 MCAN Overview ... 2600
22.1.2 MCAN Functional Description ... 2601
22.1.3 MCAN Register Manual ... 2634
22.2 Modular Controller Area Network (68xx MCAN) ... 2733
22.2.1 MCAN Overview ... 2733
22.2.2 MCAN Functional Description ... 2735
22.2.3 MCAN Register Manual ... 2768

23 Multi-Buffered Serial Peripheral Interface Module (MibSPI) ... 2871
23.1 Overview ... 2872
23.1.1 Features ... 2872
23.1.2 Pin Configurations ... 2873
23.1.3 MibSPI /SPI Configurations ... 2874
23.2 Basic Operation ... 2874
23.2.1 SPI Mode ... 2874
23.2.2 MibSPI Mode ... 2876
23.2.3 DMA Requests ... 2877
23.2.4 Interrupts ... 2879
23.2.5 Physical Interface ... 2881
23.2.6 Advanced Module Configuration Options ... 2885
23.2.7 General-Purpose I/O ... 2903
23.2.8 Low-Power Mode ... 2903
23.2.9 Safety Features ... 2903
23.2.10 Test Features ... 2905
23.2.11 Module Configuration ... 2907
23.3 Control Registers ... 2909
23.3.1 MSS_MIBSPIA Registers ... 2910
23.3.2 MSS_MIBSPIB Registers ... 3026
23.4 Multi-Buffer RAM ... 3142
23.4.1 Multi-Buffer RAM Auto Initialization ... 3143
23.4.2 Multi-Buffer RAM Register Summary ... 3143
23.4.3 Multi-Buffer RAM Transmit Data Register ... 3144
23.4.4 Multi-buffer RAM Receive Buffer Register ... 3146
23.5 Parity\ECC Memory ... 3148
23.5.1 Example of Parity Memory Organization ... 3151
23.5.2 Example of ECC Memory Organization ... 3152
23.6 MibSPI Pin Timing Parameters ... 3153
23.6.1 Master Mode Timings for SPI/MibSPI ... 3153
23.6.2 Slave Mode Timings for SPI/MibSPI ... 3155
23.6.3 Master Mode Timing Parameter Details ... 3156
23.6.4 Slave Mode Timing Parameter Details ... 3156

24 Quad Serial Peripheral Interface (QSPI) ... 3157
24.1 Quad Serial Peripheral Interface Overview ... 3158
24.2 QSPI Functional Description ... 3159
24.2.1 QSPI Block Diagram ... 3159
24.2.2 QSPI Clock Configuration ... 3164
24.2.3 QSPI Interrupt Requests ... 3164

24.2.4 QSPI Memory Regions 3166
24.3 QSPI Register Manual 3167
24.3.1 MSS_QSPI Registers 3168
25 Inter-Integrated Circuit (I2C) 3190
25.1 Overview 3191
25.1.1 Introduction to the I2C Module 3191
25.1.2 Functional Overview 3191
25.1.3 Clock Generation 3194
25.2 I2C Module Operation 3195
25.2.1 Input and Output Voltage Levels 3195
25.2.2 I2C Module Reset Conditions 3195
25.2.3 I2C Module Data Validity 3195
25.2.4 I2C Module Start and Stop Conditions 3196
25.2.5 Serial Data Formats 3196
25.2.6 NACK Bit Generation 3198
25.3 I2C Operation Modes 3199
25.3.1 Master Transmitter Mode 3199
25.3.2 Master Receiver Mode 3199
25.3.3 Slave Transmitter Mode 3199
25.3.4 Slave Receiver Mode 3199
25.3.5 Low Power Mode 3200
25.3.6 Free Run Mode 3200
25.3.7 Ignore NACK Mode 3200
25.4 I2C Module Integrity 3201
25.4.1 Arbitration 3201
25.4.2 I2C Clock Generation and Synchronization 3202
25.4.3 Prescaler 3202
25.4.4 Noise Filter 3202
25.5 Operational Information 3203
25.5.1 I2C Module Interrupts 3203
25.5.2 DMA Controller Events 3204
25.5.3 I2C Enable/Disable 3204
25.5.4 General Purpose I/O 3204
25.5.5 Pull Up/Pull Down Function 3205
25.5.6 Open Drain Function 3205
25.6 MSS_I2C Registers 3206
25.6.1 ICOAR Register (Offset = 0h) [reset = 0h] 3207
25.6.2 ICIMR Register (Offset = 4h) [reset = 0h] 3208
25.6.3 ICSTR Register (Offset = 8h) [reset = 0h] 3209
25.6.4 ICCLKL Register (Offset = Ch) [reset = 0h] 3211
25.6.5 ICCLKH Register (Offset = 10h) [reset = 0h] 3212
25.6.6 ICCNT Register (Offset = 14h) [reset = 0h] 3213
25.6.7 ICDRR Register (Offset = 18h) [reset = 0h] 3214
25.6.8 ICSAR Register (Offset = 1Ch) [reset = 0h] 3215
25.6.9 ICDXR Register (Offset = 20h) [reset = 0h] 3216
25.6.10 ICMDR Register (Offset = 24h) [reset = 0h] 3217
25.6.11 ICIVR Register (Offset = 28h) [reset = 0h] 3220
25.6.12 ICEMDR Register (Offset = 2Ch) [reset = 0h] 3221
25.6.13 ICPSC Register (Offset = 30h) [reset = 0h] 3222
25.6.14 ICPID1 Register (Offset = 34h) [reset = 0h] 3223
25.6.15 ICPID2 Register (Offset = 38h) [reset = 0h] 3224
25.6.16 ICDMAC Register (Offset = 3Ch) [reset = 0h] 3225
25.6.17 ICPFUNC Register (Offset = 48h) [reset = 0h] 3226

25.6.18 ICPDIR Register (Offset = 4Ch) [reset = 0h] ... 3227
25.6.19 ICPDIN Register (Offset = 50h) [reset = 0h] ... 3228
25.6.20 ICPDOUT Register (Offset = 54h) [reset = 0h] ... 3229
25.6.21 ICPDSET Register (Offset = 58h) [reset = 0h] ... 3230
25.6.22 ICPDCLR Register (Offset = 5Ch) [reset = 0h] ... 3231
25.6.23 ICPDRV Register (Offset = 60h) [reset = 0h] ... 3232
25.7 Sample Waveforms ... 3233
26 Serial Communication Interface (SCI) ... 3234
26.1 Introduction ... 3235
26.1.1 SCI Features ... 3235
26.1.2 Block Diagram ... 3235
26.2 SCI Communication Formats ... 3237
26.2.1 SCI Frame Formats ... 3237
26.2.2 SCI Timing Mode ... 3237
26.2.3 SCI Baud Rate ... 3238
26.2.4 SCI Multiprocessor Communication Modes ... 3239
26.3 SCI Interrupts ... 3242
26.3.1 Transmit Interrupt ... 3243
26.3.2 Receive Interrupt ... 3243
26.3.3 WakeUp Interrupt ... 3243
26.3.4 Error Interrupts ... 3244
26.4 SCI DMA Interface ... 3245
26.4.1 Receive DMA Requests ... 3245
26.4.2 Transmit DMA Requests ... 3246
26.5 SCI Configurations ... 3246
26.5.1 Receiving Data ... 3247
26.5.2 Transmitting Data ... 3247
26.6 SCI Low Power Mode ... 3248
26.6.1 Sleep Mode for Multiprocessor Communication ... 3249
26.7 SCI Control Registers ... 3250
26.7.1 SCI Global Control Register 0 (SCIGCR0) ... 3251
26.7.2 SCI Global Control Register 1 (SCIGCR1) ... 3252
26.7.3 SCI Set Interrupt Register (SCISETINT) ... 3255
26.7.4 SCI Clear Interrupt Register (SCICLEARINT) ... 3257
26.7.5 SCI Set Interrupt Level Register (SCISETINTLVL) ... 3259
26.7.6 SCI Clear Interrupt Level Register (SCICLEARINTLVL) ... 3260
26.7.7 SCI Flags Register (SCIFLR) ... 3262
26.7.8 SCI Interrupt Vector Offset 0 (SCIINTVECT0) ... 3266
26.7.9 SCI Interrupt Vector Offset 1 (SCIINTVECT1) ... 3266
26.7.10 SCI Format Control Register (SCIFORMAT) ... 3267
26.7.11 Baud Rate Selection Register (BRS) ... 3268
26.7.12 SCI Data Buffers (SCIED, SCIRD, SCITD) ... 3269
26.7.13 SCI Pin I/O Control Register 0 (SCIPIO0) ... 3270
26.7.14 SCI Pin I/O Control Register 1 (SCIPIO1) ... 3271
26.7.15 SCI Pin I/O Control Register 2 (SCIPIO2) ... 3272
26.7.16 SCI Pin I/O Control Register 3 (SCIPIO3) ... 3273
26.7.17 SCI Pin I/O Control Register 4 (SCIPIO4) ... 3274
26.7.18 SCI Pin I/O Control Register 5 (SCIPIO5) ... 3275
26.7.19 SCI Pin I/O Control Register 6 (SCIPIO6) ... 3276
26.7.20 SCI Pin I/O Control Register 7 (SCIPIO7) ... 3277
26.7.21 SCI Pin I/O Control Register 8 (SCIPIO8) ... 3277
26.7.22 Input/Output Error Enable (IODFTCTRL) Register ... 3278
26.8 GPIO Functionality ... 3280

26.8.1 GPIO Functionality 3280
26.8.2 Under Reset 3280
26.8.3 Out of Reset 3281
26.8.4 Open-Drain Feature Enabled on a Pin 3281
26.8.5 Summary 3281
27 Debug Architecture 3282
27.1 Debug Subsystem 3282
27.2 Triggering Subsystem 3283
27.2.1 Cross Trigger Input Tables 3284
28 Safety 3285
28.1 Dual Clock Comparator (DCC) 3286
28.1.1 Introduction 3286
28.1.2 Module Operation 3287
28.1.3 Clock Source Selection for Counter0 and Counter1 3291
28.1.4 DCC Registers 3292
28.2 Error Signaling Module (ESM) 3303
28.2.1 Overview 3303
28.2.2 Module Operation 3306
28.2.3 Recommended Programming Procedure 3309
28.2.4 MSS_ESM Registers 3310
28.3 Cyclic Redundancy Check (CRC) 3340
28.3.1 Overview 3340
28.3.2 Module Operation 3342
28.3.3 Example 3352
28.3.4 MSS_MCRC Registers 3356
28.4 Programmable Built-In Self-Test (PBIST) 3425
28.4.1 Overview 3425
28.4.2 RAM/ROM Grouping and Algorithm 3427
28.4.3 PBIST Registers 3429
28.5 Self-Test Controller (STC) 3441
28.5.1 Integration Spec 3441
28.5.2 General Description 3442
28.5.3 Block Diagram 3443
28.5.4 Module Description 3444
28.5.5 STC Flow 3446
28.5.6 ROM Organization 3447
28.5.7 STC Registers 3454
28.6 Core Clock Comparator (CCC) Module 3526
28.6.1 Description 3526
28.6.2 Block Diagram 3527
28.6.3 Perform Clock Comparison 3527
28.6.4 Recommended Programmation 3528
A Glossary 3529
Revision History 3549

List of Figures

1-1. 14xx Block Diagram 136
1-2. System Interconnect 139
1-3. Integration of MSS_CCCA and MSS_CCCB Modules 144
1-4. MSS_DCCA/MSS_DCCB Integration Diagram 145
1-5. Integration of MSS_DMA Module 146
1-6. Integration of MSS_RTIA and MSS_RTIB, Using the RTI Module 148
1-7. Integration Block Diagram for MSS_GIO 149
1-8. Integration Block Diagram for MSS_DCAN Module 152
1-9. MSS_MIBSPIA Integration 153
1-10. MSS_SPIB Integration Diagram 154
1-11. EDMA Controller Integration 155
1-12. 14xx MSS_ESM Integration Diagram 157
2-1. 16xx Block Diagram 163
2-2. System Interconnect 167
2-3. Integration of MSS_CCCA and MSS_CCCB Modules 174
2-4. MSS_CCCB Integration to MSS_WD 175
2-5. MSS_DCCA/MSS_DCCB Integration Diagram 176
2-6. Integration of MSS_DMA and MSS_DMA2 Module 180
2-7. Integration of MSS_RTIA and MSS_RTIB, WDT Using the MSS_RTIB Module 183
2-8. Integration Block Diagram for MSS_GIO 184
2-9. MSS_DMM Integration 185
2-10. Multiple MSS_ETPWM Modules 186
2-11. Integration Block Diagram for MSS_DCAN Module 190
2-12. MSS_MIBSPIA Integration 191
2-13. MSS_MIBSPIB Integration 192
2-14. EDMA Controller Integration (1 of 2) 194
2-15. EDMA Controller Integration (2 of 2) 195
2-16. 16xx MSS_ESM/DSS_ESM Integration Diagram 199
2-17. 18xx Block Diagram 204
2-18. System Interconnect 208
2-19. Integration of MSS_CCCA and MSS_CCCB Modules 216
2-20. MSS_CCCB Integration to MSS_WD 217
2-21. MSS_DCCA/MSS_DCCB Integration Diagram 218
2-22. Integration of MSS_DMA and MSS_DMA2 Module 222
2-23. Integration of MSS_RTIA and MSS_RTIB, WDT Using the MSS_RTIB Module 225
2-24. Integration Block Diagram for MSS_GIO 226
2-25. MSS_DMM Integration 227
2-26. Multiple MSS_ETPWM Modules 228
2-27. Integration Block Diagram for MSS_DCAN Module 232
2-28. MSS_MIBSPIA Integration 233
2-29. MSS_MIBSPIB Integration 234
2-30. EDMA Controller Integration (1 of 2) 236
2-31. EDMA Controller Integration (2 of 2) 237
2-32. 18xx MSS_ESM/DSS_ESM Integration Diagram 241
3-1. 68xx Block Diagram 248
3-2. System Interconnect 252
3-3. Integration of MSS_CCCA and MSS_CCCB Modules 261

3-4. MSS_DCCA/MSS_DCCB Integration Diagram 262
3-5. Integration of MSS_DMA and MSS_DMA2 Module 266
3-6. Integration of MSS_RTIA and MSS_RTIB, WDT Using the MSS_RTIB Module 269
3-7. Integration Block Diagram for MSS_GIO 270
3-8. MSS_DMM Integration 271
3-9. Multiple MSS_ETPWM Modules 272
3-10. Integration Block Diagram for Module Controller Area Network (MSS_MCAN) 276
3-11. MSS_MIBSPIA Integration 277
3-12. MSS_MIBSPIB Integration 278
3-13. EDMA Controller Integration (1 of 2) 280
3-14. EDMA Controller Integration (2 of 2) 281
3-15. 68xx MSS_ESM/DSS_ESM Integration Diagram 285
4-1. Clock Overview 291
4-2. Reset Overview 294
4-3. Power Domains 297
4-4. Power Domain Controls 297
4-5. BSSCTL Register 304
4-6. EXTCLKDIV Register 305
4-7. EXTCLKSRCSEL Register 306
4-8. EXTCLKCTL Register 307
4-9. SOFTSYSRST Register 308
4-10. WDRSTEN Register 309
4-11. SYSRSTCAUSE Register 310
4-12. SYSRSTCAUSECLR Register 311
4-13. MISCCAPT Register 312
4-14. DCDCCTL0 Register 313
4-15. DCDCCTL1 Register 314
4-16. MISCCTL Register 315
4-17. USERMODEEN Register 316
4-18. LVDSPADCTL0 Register 317
4-19. LVDSPADCTL1 Register 318
4-20. DFTREG0 Register 319
4-21. DFTREG1 Register 320
4-22. DFTREG2 Register 321
4-23. DFTREG3 Register 322
4-24. DFTREG4 Register 323
4-25. DFTREG5 Register 324
4-26. SHMEMINITADDR Register 325
4-27. SHMEMINITECC Register 326
4-28. DSSMEMBANKEN Register 327
4-29. DSSMEMTAB0 Register 328
4-30. DSSMEMTAB1 Register 329
4-31. TCMAMEMBANK_EN Register 330
4-32. TCMAMEMTAB0 Register 331
4-33. TCMAMEMTAB1 Register 332
4-34. TCMBMEMBANKEN Register 333
4-35. TCMBMEMTAB0 Register 334
4-36. TCMBMEM_TAB1 Register 335
4-37. MEMINITSTART Register 336

4-38. MEMINITDONE Register 337
4-39. MSS_SIGNATURE Register 338
4-40. MISCCTL1 Register 339
4-41. USERMODEEN2 Register 340
4-42. SYSTICK Register 341
4-43. SOFTRST2 Register 344
4-44. CLKDIVCTL0 Register 345
4-45. CLKSRCSEL0 Register 346
4-46. CR4CTL Register 347
4-47. CLKGATE Register 348
4-48. CLKSRCSEL1 Register 349
4-49. CURRCLKDIV0 Register 350
4-50. MEMINITSTART Register 351
4-51. CURRCLKDIV1 Register 352
4-52. MEMINITDONE Register 353
4-53. USERMODEEN Register 354
4-54. NSYSPERUSERMODEN Register 355
4-55. ESMGATE0 Register 356
4-56. ESMGATE1 Register 357
4-57. ESMGATE2 Register 358
4-58. ESMGATE3 Register 359
4-59. ESMGATE4 Register 360
4-60. KEY Register 361
4-61. SWIRQA Register 362
4-62. SWIRQB Register 363
4-63. MISCCTL0 Register 364
4-64. ATCMERRCAPTCTL Register 365
4-65. B0TCMERRCAPTCTL Register 366
4-66. B1TCMERRCAPTCTL Register 367
4-67. SOFTCORERST Register 368
4-68. RSTCAUSE Register 369
4-69. RSTCAUSECLR Register 370
4-70. SPITRIGSRC Register 371
4-71. CLKINUSE Register 372
4-72. ECCEN Register 373
4-73. ECCCAPT Register 374
4-74. CLKDIVCTL2 Register 375
4-75. SWIRQC Register 376
4-76. GPCFG0 Register 378
4-77. GPCFG1 Register 379
4-78. GPCFG2 Register 380
4-79. GPCFG3 Register 381
4-80. GPCFG4 Register 382
4-81. CCCACFG0 Register 383
4-82. CCCACFG1 Register 384
4-83. CCCACFG2 Register 385
4-84. CCCACFG3 Register 386
4-85. CCCBCFG0 Register 387
4-86. CCCBCFG1 Register 388

4-87. CCCBCFG2 Register 389
4-88. CCCBCFG3 Register 390
4-89. CCCACNTVAL Register 391
4-90. CCCBCNTVAL Register 392
4-91. CCCABERRSTAT Register 393
4-92. USERMODEEN Register 394
4-93. RTIEVENTCAPTURESEL Register 398
4-94. ADCBUFCFG1 Register 399
4-95. ADCBUFCFG2 Register 400
4-96. ADCBUFCFG3 Register 401
4-97. ADCBUFCFG4 Register 402
4-98. CQCFG1 Register 403
4-99. TPCCPARSTATCFG Register 404
4-100. CSI2TXPARSTATCFG Register 405
4-101. CSICFG1 Register 406
4-102. TPTC0WRMPUSTADD0 Register 407
4-103. TPTC0WRMPUSTADD1 Register 408
4-104. TPTC0WRMPUSTADD2 Register 409
4-105. TPTC0WRMPUSTADD3 Register 410
4-106. TPTC0WRMPUSTADD4 Register 411
4-107. TPTC0WRMPUSTADD5 Register 412
4-108. TPTC0WRMPUSTADD6 Register 413
4-109. TPTC0WRMPUSTADD7 Register 414
4-110. TPTC0WRMPUENDADD0 Register 415
4-111. TPTC0WRMPUENDADD1 Register 416
4-112. TPTC0WRMPUENDADD2 Register 417
4-113. TPTC0WRMPUENDADD3 Register 418
4-114. TPTC0WRMPUENDADD4 Register 419
4-115. TPTC0WRMPUENDADD5 Register 420
4-116. TPTC0WRMPUENDADD6 Register 421
4-117. TPTC0WRMPUENDADD7 Register 422
4-118. TPTC0WRMPUERRADD Register 423
4-119. TPTC0RDMPUSTADD0 Register 424
4-120. TPTC0RDMPUSTADD1 Register 425
4-121. TPTC0RDMPUSTADD2 Register 426
4-122. TPTC0RDMPUSTADD3 Register 427
4-123. TPTC0RDMPUSTADD4 Register 428
4-124. TPTC0RDMPUSTADD5 Register 429
4-125. TPTC0RDMPUSTADD6 Register 430
4-126. TPTC0RDMPUSTADD7 Register 431
4-127. TPTC0RDMPUENDADD0 Register 432
4-128. TPTC0RDMPUENDADD1 Register 433
4-129. TPTC0RDMPUENDADD2 Register 434
4-130. TPTC0RDMPUENDADD3 Register 435
4-131. TPTC0RDMPUENDADD4 Register 436
4-132. TPTC0RDMPUENDADD5 Register 437
4-133. TPTC0RDMPUENDADD6 Register 438
4-134. TPTC0RDMPUENDADD7 Register 439
4-135. TPTC0RDMPUERRADD Register 440

4-136. TPTC1WRMPUSTADD0 Register 441
4-137. TPTC1WRMPUSTADD1 Register 442
4-138. TPTC1WRMPUSTADD2 Register 443
4-139. TPTC1WRMPUSTADD3 Register 444
4-140. TPTC1WRMPUSTADD4 Register 445
4-141. TPTC1WRMPUSTADD5 Register 446
4-142. TPTC1WRMPUSTADD6 Register 447
4-143. TPTC1WRMPUSTADD7 Register 448
4-144. TPTC1WRMPUENDADD0 Register 449
4-145. TPTC1WRMPUENDADD1 Register 450
4-146. TPTC1WRMPUENDADD2 Register 451
4-147. TPTC1WRMPUENDADD3 Register 452
4-148. TPTC1WRMPUENDADD4 Register 453
4-149. TPTC1WRMPUENDADD5 Register 454
4-150. TPTC1WRMPUENDADD6 Register 455
4-151. TPTC1WRMPUENDADD7 Register 456
4-152. TPTC1WRMPUERRADD Register 457
4-153. TPTC1RDMPUSTADD0 Register 458
4-154. TPTC1RDMPUSTADD1 Register 459
4-155. TPTC1RDMPUSTADD2 Register 460
4-156. TPTC1RDMPUSTADD3 Register 461
4-157. TPTC1RDMPUSTADD4 Register 462
4-158. TPTC1RDMPUSTADD5 Register 463
4-159. TPTC1RDMPUSTADD6 Register 464
4-160. TPTC1RDMPUSTADD7 Register 465
4-161. TPTC1RDMPUENDADD0 Register 466
4-162. TPTC1RDMPUENDADD1 Register 467
4-163. TPTC1RDMPUENDADD2 Register 468
4-164. TPTC1RDMPUENDADD3 Register 469
4-165. TPTC1RDMPUENDADD4 Register 470
4-166. TPTC1RDMPUENDADD5 Register 471
4-167. TPTC1RDMPUENDADD6 Register 472
4-168. TPTC1RDMPUENDADD7 Register 473
4-169. TPTC1RDMPUERRADD Register 474
4-170. TPTCMPUVALIDCFG Register 475
4-171. TPTCMPUENCFG Register 476
4-172. TESTPATTERNRX1ICFG Register 477
4-173. TESTPATTERNRX2ICFG Register 478
4-174. TESTPATTERNRX3ICFG Register 479
4-175. TESTPATTERNRX4ICFG Register 480
4-176. TESTPATTERNRX1QCFG Register 481
4-177. TESTPATTERNRX2QCFG Register 482
4-178. TESTPATTERNRX3QCFG Register 483
4-179. TESTPATTERNRX4QCFG Register 484
4-180. TESTPATTERNVLDCFG Register 485
4-181. DSSMISC Register 486
4-182. DSSMISC2 Register 487
4-183. BSSCTL Register 490
4-184. DSSCTL Register 491

4-185. EXTCLKDIV Register 492
4-186. EXTCLKSRCSEL Register 493
4-187. EXTCLKCTL Register 494
4-188. SOFTSYSRST Register 495
4-189. WDRSTEN Register 496
4-190. SYSRSTCAUSE Register 497
4-191. SYSRSTCAUSECLR Register 498
4-192. MISCCAPT Register 499
4-193. DCDCCTL0 Register 500
4-194. DCDCCTL1 Register 501
4-195. USERMODEEN Register 502
4-196. LVDSPADCTL0 Register 503
4-197. LVDSPADCTL1 Register 504
4-198. DFTREG0 Register 505
4-199. DFTREG1 Register 506
4-200. DFTREG5 Register 507
4-201. MEMINITDONE Register 508
4-202. SPARE0_3 Register 509
4-203. MSS_SIGNATURE Register 510
4-204. GEMBOOTSTCEN Register 511
4-205. MISCCTL1 Register 512
4-206. USERMODEEN2 Register 513
4-207. SYSTICK Register 514
4-208. SECURECFGREG1 Register 515
4-209. SECURECFGREG2 Register 516
4-210. SECURECFGREG3 Register 517
4-211. SECURECFGREG4 Register 518
4-212. SECURERAMREG Register 519
4-213. SPAREMULTIBIT Register 520
4-214. SPARE4_9 Register 521
4-215. UID31TO0 Register 522
4-216. UID63TO32 Register 523
4-217. UID95TO64 Register 524
4-218. UID119TO96 Register 525
4-219. MEMINITSTARTSHMEM Register 526
4-220. MEMINITDONESHMEM Register 527
4-221. DSSMEMTAB0 Register 528
4-222. TCMAMEMTAB Register 529
4-223. TCMBMEMTAB Register 530
4-224. SHMEMBANKSEL3TO0 Register 531
4-225. SHMEMBANKSEL7TO4 Register 532
4-226. PBISTCLKCTL Register 533
4-227. SOFTRST1 Register 536
4-228. SOFTRST2 Register 537
4-229. CLKDIVCTL0 Register 538
4-230. CLKSRCSEL0 Register 539
4-231. CR4CTL Register 540
4-232. CLKGATE Register 541
4-233. CLKSRCSEL1 Register 542

4-234. CURRCLKDIV0 Register 543
4-235. RTICURRCLKDIV Register 544
4-236. MEMINITSTART Register 545
4-237. CURRCLKDIV1 Register 546
4-238. MEMINITDONE Register 547
4-239. ECCENMSSGEM Register 548
4-240. ECCCAPTMSSGEM Register 549
4-241. ECCENBSSGEM Register 550
4-242. ECCCAPTBSSGEM Register 551
4-243. USERMODEEN Register 552
4-244. NSYSPERUSERMODEN Register 553
4-245. SECURERAMMMI Register 554
4-246. SECURERAMECC Register 555
4-247. ESMGATE0 Register 556
4-248. ESMGATE1 Register 557
4-249. ESMGATE2 Register 558
4-250. ESMGATE3 Register 559
4-251. ESMGATE4 Register 560
4-252. KEY Register 561
4-253. SWIRQA Register 562
4-254. SWIRQB Register 563
4-255. MISCCTL0 Register 564
4-256. ATCMERRCAPTCTL Register 565
4-257. B0TCMERRCAPTCTL Register 566
4-258. B1TCMERRCAPTCTL Register 567
4-259. SOFTCORERST Register 568
4-260. RSTCAUSE Register 569
4-261. RSTCAUSECLR Register 570
4-262. SPITRIGSRC Register 571
4-263. CLKINUSE Register 572
4-264. ECCENMSSBSS Register 573
4-265. ECCCAPTMSSBSS Register 574
4-266. CLKDIVCTL2 Register 575
4-267. SWIRQC Register 576
4-268. GPCFG0 Register 579
4-269. GPCFG1 Register 580
4-270. GPCFG2 Register 581
4-271. GPCFG3 Register 582
4-272. GPCFG4 Register 583
4-273. GPCFG6 Register 584
4-274. GPCFG11 Register 585
4-275. CCCACFG0 Register 586
4-276. CCCACFG1 Register 587
4-277. CCCACFG2 Register 588
4-278. CCCACFG3 Register 589
4-279. CCCBCFG0 Register 590
4-280. CCCBCFG1 Register 591
4-281. CCCBCFG2 Register 592
4-282. CCCBCFG3 Register 593

4-283. CCCACNTVAL Register 594
4-284. CCCBCNTVAL Register 595
4-285. CCCABERRSTAT Register 596
4-286. USERMODEEN Register 597
4-287. EPWMCFG Register 598
4-288. DMMSWINT0 Register 599
4-289. DMMSWINT1 Register 600
4-290. DMMSWINTSEL0 Register 601
4-291. DMMSWINTSEL1 Register 602
4-292. CCCBWDEN Register 603
4-293. GPIOINTREDGESEL Register 604
4-294. PWMDMATRIGEN Register 605
4-295. JTAGTXDATA Register 606
4-296. JTAGTXCONTROL Register 607
4-297. JTAGRXDATA Register 608
4-298. JTAGTXRXACK Register 609
4-299. JTAGRXCONTROL Register 610
4-300. MSS2GEMSWIRQ Register 611
4-301. CSETBFLUSH Register 612
4-302. RTIEVENTCAPTURESEL Register 616
4-303. CQCFG1 Register 617
4-304. TPCCPARSTATCFG Register 618
4-305. TPTC0WRMPUSTADD0 Register 619
4-306. TPTC0WRMPUSTADD1 Register 620
4-307. TPTC0WRMPUSTADD2 Register 621
4-308. TPTC0WRMPUSTADD3 Register 622
4-309. TPTC0WRMPUSTADD4 Register 623
4-310. TPTC0WRMPUSTADD5 Register 624
4-311. TPTC0WRMPUENDADD0 Register 625
4-312. TPTC0WRMPUENDADD1 Register 626
4-313. TPTC0WRMPUENDADD2 Register 627
4-314. TPTC0WRMPUENDADD3 Register 628
4-315. TPTC0WRMPUENDADD4 Register 629
4-316. TPTC0WRMPUENDADD5 Register 630
4-317. TPTC0WRMPUERRADD Register 631
4-318. TPTC0RDMPUSTADD0 Register 632
4-319. TPTC0RDMPUSTADD1 Register 633
4-320. TPTC0RDMPUSTADD2 Register 634
4-321. TPTC0RDMPUSTADD3 Register 635
4-322. TPTC0RDMPUSTADD4 Register 636
4-323. TPTC0RDMPUSTADD5 Register 637
4-324. TPTC0RDMPUENDADD0 Register 638
4-325. TPTC0RDMPUENDADD1 Register 639
4-326. TPTC0RDMPUENDADD2 Register 640
4-327. TPTC0RDMPUENDADD3 Register 641
4-328. TPTC0RDMPUENDADD4 Register 642
4-329. TPTC0RDMPUENDADD5 Register 643
4-330. TPTC0RDMPUERRADD Register 644
4-331. TPTC1WRMPUSTADD0 Register 645

4-332. TPTC1WRMPUSTADD1 Register 646
4-333. TPTC1WRMPUSTADD2 Register 647
4-334. TPTC1WRMPUSTADD3 Register 648
4-335. TPTC1WRMPUSTADD4 Register 649
4-336. TPTC1WRMPUSTADD5 Register 650
4-337. TPTC1WRMPUENDADD0 Register 651
4-338. TPTC1WRMPUENDADD1 Register 652
4-339. TPTC1WRMPUENDADD2 Register 653
4-340. TPTC1WRMPUENDADD3 Register 654
4-341. TPTC1WRMPUENDADD4 Register 655
4-342. TPTC1WRMPUENDADD5 Register 656
4-343. TPTC1WRMPUERRADD Register 657
4-344. TPTC1RDMPUSTADD0 Register 658
4-345. TPTC1RDMPUSTADD1 Register 659
4-346. TPTC1RDMPUSTADD2 Register 660
4-347. TPTC1RDMPUSTADD3 Register 661
4-348. TPTC1RDMPUSTADD4 Register 662
4-349. TPTC1RDMPUSTADD5 Register 663
4-350. TPTC1RDMPUENDADD0 Register 664
4-351. TPTC1RDMPUENDADD1 Register 665
4-352. TPTC1RDMPUENDADD2 Register 666
4-353. TPTC1RDMPUENDADD3 Register 667
4-354. TPTC1RDMPUENDADD4 Register 668
4-355. TPTC1RDMPUENDADD5 Register 669
4-356. TPTC1RDMPUERRADD Register 670
4-357. TPTCMPUVALIDCFG Register 671
4-358. TPTCMPUENCFG Register 672
4-359. TESTPATTERNRX1ICFG Register 673
4-360. TESTPATTERNRX2ICFG Register 674
4-361. TESTPATTERNRX3ICFG Register 675
4-362. TESTPATTERNRX4ICFG Register 676
4-363. TESTPATTERNRX1QCFG Register 677
4-364. TESTPATTERNRX2QCFG Register 678
4-365. TESTPATTERNRX3QCFG Register 679
4-366. TESTPATTERNRX4QCFG Register 680
4-367. TESTPATTERNVLDCFG Register 681
4-368. TPCC1PARSTATCFG Register 682
4-369. DMMSWINT1 Register 683
4-370. DSSINTRCFG Register 684
4-371. MPUMSTIDCFG1 Register 685
4-372. MPUMSTIDCFG2 Register 686
4-373. MPUMSTIDCFG3 Register 687
4-374. HSRAM1ECCCFG Register 688
4-375. DATATRRAMECCCFG Register 689
4-376. ADCBUFPINGECCCFG Register 690
4-377. ADCBUFPONGECCCFG Register 691
4-378. UMAP0PARITYCFG1 Register 692
4-379. UMAP0PARITYCFG2 Register 693
4-380. UMAP0PARITYCFG3 Register 694

4-381. UMAP1PARITYCFG1 Register 695
4-382. UMAP1PARITYCFG2 Register 696
4-383. UMAP1PARITYCFG3 Register 697
4-384. ESMGRP2MASKCFG Register 698
4-385. L2MEMINITCFG1 Register 699
4-386. L2MEMINITCFG2 Register 701
4-387. GEMRSTCAUSE Register 702
4-388. GEMPWRSMCFG4 Register 703
4-389. PWRSMWAKEMASK0 Register 704
4-390. PWRSMWAKEMASK1 Register 705
4-391. PWRSMWAKEMASK2 Register 706
4-392. PWRSMMISEVTMASK0 Register 707
4-393. PWRSMMISEVTMASK1 Register 708
4-394. PWRSMMISEVTMASK2 Register 709
4-395. PWRSMWAKESRCSTAT0 Register 710
4-396. PWRSMWAKESRCSTAT1 Register 711
4-397. PWRSMWAKESRCSTAT2 Register 712
4-398. PWRSMEVNTMONSTAT0 Register 713
4-399. PWRSMEVNTMONSTAT1 Register 714
4-400. PWRSMEVNTMONSTAT2 Register 715
4-401. PWRSMWAKESRCSTATCLR0 Register 716
4-402. PWRSMWAKESRCSTATCLR1 Register 717
4-403. PWRSMWAKESRCSTATCLR2 Register 718
4-404. ADCBUFCFG1 Register 719
4-405. ADCBUFCFG2 Register 720
4-406. ADCBUFCFG3 Register 721
4-407. ADCBUFCFG4 Register 722
4-408. STCPBISTSMCFG1 Register 723
4-409. STCPBISTSMCFG2 Register 724
4-410. RTI2EVENTCAPTURESEL Register 725
4-411. DSSMISC5 Register 726
4-412. TPTC2WRMPUSTADD0 Register 729
4-413. TPTC2WRMPUSTADD1 Register 730
4-414. TPTC2WRMPUSTADD2 Register 731
4-415. TPTC2WRMPUSTADD3 Register 732
4-416. TPTC2WRMPUSTADD4 Register 733
4-417. TPTC2WRMPUSTADD5 Register 734
4-418. TPTC2WRMPUENDADD0 Register 735
4-419. TPTC2WRMPUENDADD1 Register 736
4-420. TPTC2WRMPUENDADD2 Register 737
4-421. TPTC2WRMPUENDADD3 Register 738
4-422. TPTC2WRMPUENDADD4 Register 739
4-423. TPTC2WRMPUENDADD5 Register 740
4-424. TPTC2WRMPUERRADD Register 741
4-425. TPTC2RDMPUSTADD0 Register 742
4-426. TPTC2RDMPUSTADD1 Register 743
4-427. TPTC2RDMPUSTADD2 Register 744
4-428. TPTC2RDMPUSTADD3 Register 745
4-429. TPTC2RDMPUSTADD4 Register 746

4-430. TPTC2RDMPUSTADD5 Register 747
4-431. TPTC2RDMPUENDADD0 Register 748
4-432. TPTC2RDMPUENDADD1 Register 749
4-433. TPTC2RDMPUENDADD2 Register 750
4-434. TPTC2RDMPUENDADD3 Register 751
4-435. TPTC2RDMPUENDADD4 Register 752
4-436. TPTC2RDMPUENDADD5 Register 753
4-437. TPTC2RDMPUERRADD Register 754
4-438. TPTC3WRMPUSTADD0 Register 755
4-439. TPTC3WRMPUSTADD1 Register 756
4-440. TPTC3WRMPUSTADD2 Register 757
4-441. TPTC3WRMPUSTADD3 Register 758
4-442. TPTC3WRMPUSTADD4 Register 759
4-443. TPTC3WRMPUSTADD5 Register 760
4-444. TPTC3WRMPUENDADD0 Register 761
4-445. TPTC3WRMPUENDADD1 Register 762
4-446. TPTC3WRMPUENDADD2 Register 763
4-447. TPTC3WRMPUENDADD3 Register 764
4-448. TPTC3WRMPUENDADD4 Register 765
4-449. TPTC3WRMPUENDADD5 Register 766
4-450. TPTC3WRMPUERRADD Register 767
4-451. TPTC3RDMPUSTADD0 Register 768
4-452. TPTC3RDMPUSTADD1 Register 769
4-453. TPTC3RDMPUSTADD2 Register 770
4-454. TPTC3RDMPUSTADD3 Register 771
4-455. TPTC3RDMPUSTADD4 Register 772
4-456. TPTC3RDMPUSTADD5 Register 773
4-457. TPTC3RDMPUENDADD0 Register 774
4-458. TPTC3RDMPUENDADD1 Register 775
4-459. TPTC3RDMPUENDADD2 Register 776
4-460. TPTC3RDMPUENDADD3 Register 777
4-461. TPTC3RDMPUENDADD4 Register 778
4-462. TPTC3RDMPUENDADD5 Register 779
4-463. TPTC3RDMPUERRADD Register 780
4-464. TPTCMPUVALIDCFG2 Register 781
4-465. TPTCMPUENCFG2 Register 782
4-466. L3ECCCFG1 Register 783
4-467. L3ECCCFG2 Register 784
4-468. DSS2MSSSWIRQ Register 785
4-469. BSSCTL Register 789
4-470. DSSCTL Register 790
4-471. EXTCLKDIV Register 791
4-472. EXTCLKSRCSEL Register 792
4-473. EXTCLKCTL Register 793
4-474. SOFTSYSRST Register 794
4-475. WDRSTEN Register 795
4-476. SYSRSTCAUSE Register 796
4-477. SYSRSTCAUSECLR Register 797
4-478. MISCCAPT Register 798

4-479. DCDCCTL0 Register 799
4-480. DCDCCTL1 Register 800
4-481. USERMODEEN Register 801
4-482. LVDSPADCTL0 Register 802
4-483. LVDSPADCTL1 Register 803
4-484. DFTREG0 Register 804
4-485. DFTREG1 Register 805
4-486. DFTREG5 Register 806
4-487. MEMINITDONE Register 807
4-488. SPARE0_3 Register 808
4-489. MSS_SIGNATURE Register 809
4-490. GEMBOOTSTCEN Register 810
4-491. MISCCTL1 Register 811
4-492. USERMODEEN2 Register 812
4-493. SYSTICK Register 813
4-494. SECURECFGREG1 Register 814
4-495. SECURECFGREG2 Register 815
4-496. SECURECFGREG3 Register 816
4-497. SECURECFGREG4 Register 817
4-498. SECURERAMREG Register 818
4-499. SPAREMULTIBIT Register 819
4-500. SPARE4_9 Register 820
4-501. UID31TO0 Register 821
4-502. UID63TO32 Register 822
4-503. UID95TO64 Register 823
4-504. UID119TO96 Register 824
4-505. MEMINITSTARTSHMEM Register 825
4-506. MEMINITDONESHMEM Register 826
4-507. DSSMEMTAB0 Register 827
4-508. TCMAMEMTAB Register 828
4-509. TCMBMEMTAB Register 829
4-510. SHMEMBANKSEL3TO0 Register 830
4-511. SHMEMBANKSEL7TO4 Register 831
4-512. PBISTCLKCTL Register 832
4-513. SOFTRST0 Register 835
4-514. SOFTRST1 Register 836
4-515. SOFTRST2 Register 837
4-516. SOFTRST3 Register 838
4-517. MCUIFSOFTRST0 Register 839
4-518. ECUIFSOFTRST0 Register 840
4-519. CLKDIVCTL0 Register 841
4-520. CLKSRCSEL0 Register 842
4-521. CR4CTL Register 843
4-522. SOFTRST4 Register 844
4-523. CLKGATE Register 845
4-524. CLKSRCSEL1 Register 846
4-525. CLKDIVCTL1 Register 847
4-526. CURRCLKDIV0 Register 848
4-527. RTICURRCLKDIV Register 849

4-528. MEMINITSTART Register 850
4-529. CURRCLKDIV1 Register 851
4-530. MEMINITDONE Register 852
4-531. ECCENMSSGEM Register 853
4-532. ECCCAPTMSSGEM Register 854
4-533. ECCENBSSGEM Register 855
4-534. ECCCAPTBSSGEM Register 856
4-535. USERMODEEN Register 857
4-536. NSYSPERUSERMODEN Register 858
4-537. SECURERAMMMI Register 859
4-538. SECURERAMECC Register 860
4-539. ESMGATE0 Register 861
4-540. ESMGATE1 Register 862
4-541. ESMGATE2 Register 863
4-542. ESMGATE3 Register 864
4-543. ESMGATE4 Register 865
4-544. MSSECOSPARE Register 866
4-545. KEY Register 867
4-546. DBGACKCTL0 Register 868
4-547. DBGACKCTL1 Register 869
4-548. SWIRQA Register 870
4-549. SWIRQB Register 871
4-550. MISCCTL0 Register 872
4-551. ATCMERRCAPTCTL Register 873
4-552. B0TCMERRCAPTCTL Register 874
4-553. B1TCMERRCAPTCTL Register 875
4-554. SOFTCORERST Register 876
4-555. WDOGRSTEN Register 877
4-556. RSTCAUSE Register 878
4-557. RSTCAUSECLR Register 879
4-558. SPITRIGSRC Register 880
4-559. CLKINUSE Register 881
4-560. ECCENMSSBSS Register 882
4-561. ECCCAPTMSSBSS Register 883
4-562. CLKDIVCTL2 Register 884
4-563. QSPIMCONNECT Register 885
4-564. QSPISCONNECT Register 886
4-565. SWIRQC Register 887
4-566. GPCFG0 Register 890
4-567. GPCFG1 Register 891
4-568. GPCFG2 Register 892
4-569. GPCFG3 Register 893
4-570. GPCFG4 Register 894
4-571. GPCFG6 Register 895
4-572. GPCFG11 Register 896
4-573. CCCACFG0 Register 897
4-574. CCCACFG1 Register 898
4-575. CCCACFG2 Register 899
4-576. CCCACFG3 Register 900

4-577. CCCBCFG0 Register 901
4-578. CCCBCFG1 Register 902
4-579. CCCBCFG2 Register 903
4-580. CCCBCFG3 Register 904
4-581. CCCACNTVAL Register 905
4-582. CCCBCNTVAL Register 906
4-583. CCCABERRSTAT Register 907
4-584. USERMODEEN Register 908
4-585. EPWMCFG Register 909
4-586. DMMSWINT0 Register 910
4-587. DMMSWINT1 Register 911
4-588. DMMSWINTSEL0 Register 912
4-589. DMMSWINTSEL1 Register 913
4-590. CCCBWDEN Register 914
4-591. GPIOINTREDGESEL Register 915
4-592. PWMDMATRIGEN Register 916
4-593. JTAGTXDATA Register 917
4-594. JTAGTXCONTROL Register 918
4-595. JTAGRXDATA Register 919
4-596. JTAGTXRXACK Register 920
4-597. JTAGRXCONTROL Register 921
4-598. MSS2GEMSWIRQ Register 922
4-599. CSETBFLUSH Register 923
4-600. RTIEVENTCAPTURESEL Register 927
4-601. CQCFG1 Register 928
4-602. TPCCPARSTATCFG Register 929
4-603. TPTC0WRMPUSTADD0 Register 930
4-604. TPTC0WRMPUSTADD1 Register 931
4-605. TPTC0WRMPUSTADD2 Register 932
4-606. TPTC0WRMPUSTADD3 Register 933
4-607. TPTC0WRMPUSTADD4 Register 934
4-608. TPTC0WRMPUSTADD5 Register 935
4-609. TPTC0WRMPUENDADD0 Register 936
4-610. TPTC0WRMPUENDADD1 Register 937
4-611. TPTC0WRMPUENDADD2 Register 938
4-612. TPTC0WRMPUENDADD3 Register 939
4-613. TPTC0WRMPUENDADD4 Register 940
4-614. TPTC0WRMPUENDADD5 Register 941
4-615. TPTC0WRMPUERRADD Register 942
4-616. TPTC0RDMPUSTADD0 Register 943
4-617. TPTC0RDMPUSTADD1 Register 944
4-618. TPTC0RDMPUSTADD2 Register 945
4-619. TPTC0RDMPUSTADD3 Register 946
4-620. TPTC0RDMPUSTADD4 Register 947
4-621. TPTC0RDMPUSTADD5 Register 948
4-622. TPTC0RDMPUENDADD0 Register 949
4-623. TPTC0RDMPUENDADD1 Register 950
4-624. TPTC0RDMPUENDADD2 Register 951
4-625. TPTC0RDMPUENDADD3 Register 952

4-626. TPTC0RDMPUENDADD4 Register 953
4-627. TPTC0RDMPUENDADD5 Register 954
4-628. TPTC0RDMPUERRADD Register 955
4-629. TPTC1WRMPUSTADD0 Register 956
4-630. TPTC1WRMPUSTADD1 Register 957
4-631. TPTC1WRMPUSTADD2 Register 958
4-632. TPTC1WRMPUSTADD3 Register 959
4-633. TPTC1WRMPUSTADD4 Register 960
4-634. TPTC1WRMPUSTADD5 Register 961
4-635. TPTC1WRMPUENDADD0 Register 962
4-636. TPTC1WRMPUENDADD1 Register 963
4-637. TPTC1WRMPUENDADD2 Register 964
4-638. TPTC1WRMPUENDADD3 Register 965
4-639. TPTC1WRMPUENDADD4 Register 966
4-640. TPTC1WRMPUENDADD5 Register 967
4-641. TPTC1WRMPUERRADD Register 968
4-642. TPTC1RDMPUSTADD0 Register 969
4-643. TPTC1RDMPUSTADD1 Register 970
4-644. TPTC1RDMPUSTADD2 Register 971
4-645. TPTC1RDMPUSTADD3 Register 972
4-646. TPTC1RDMPUSTADD4 Register 973
4-647. TPTC1RDMPUSTADD5 Register 974
4-648. TPTC1RDMPUENDADD0 Register 975
4-649. TPTC1RDMPUENDADD1 Register 976
4-650. TPTC1RDMPUENDADD2 Register 977
4-651. TPTC1RDMPUENDADD3 Register 978
4-652. TPTC1RDMPUENDADD4 Register 979
4-653. TPTC1RDMPUENDADD5 Register 980
4-654. TPTC1RDMPUERRADD Register 981
4-655. TPTCMPUVALIDCFG Register 982
4-656. TPTCMPUENCFG Register 983
4-657. TESTPATTERNRX1ICFG Register 984
4-658. TESTPATTERNRX2ICFG Register 985
4-659. TESTPATTERNRX3ICFG Register 986
4-660. TESTPATTERNRX4ICFG Register 987
4-661. TESTPATTERNRX1QCFG Register 988
4-662. TESTPATTERNRX2QCFG Register 989
4-663. TESTPATTERNRX3QCFG Register 990
4-664. TESTPATTERNRX4QCFG Register 991
4-665. TESTPATTERNVLDCFG Register 992
4-666. TPCC1PARSTATCFG Register 993
4-667. DMMSWINT1 Register 994
4-668. DSSINTRCFG Register 995
4-669. MPUMSTIDCFG1 Register 996
4-670. MPUMSTIDCFG2 Register 997
4-671. MPUMSTIDCFG3 Register 998
4-672. HSRAM1ECCCFG Register 999
4-673. DATATRRAMECCCFG Register 1000
4-674. ADCBUFPINGECCCFG Register 1001

4-675. ADCBUFPONGECCCFG Register 1002
4-676. UMAP0PARITYCFG1 Register 1003
4-677. UMAP0PARITYCFG2 Register 1004
4-678. UMAP0PARITYCFG3 Register 1005
4-679. UMAP1PARITYCFG1 Register 1006
4-680. UMAP1PARITYCFG2 Register 1007
4-681. UMAP1PARITYCFG3 Register 1008
4-682. ESMGRP2MASKCFG Register 1009
4-683. L2MEMINITCFG1 Register 1010
4-684. L2MEMINITCFG2 Register 1012
4-685. GEMRSTCAUSE Register 1013
4-686. GEMPWRSMCFG4 Register 1014
4-687. PWRSMWAKEMASK0 Register 1015
4-688. PWRSMWAKEMASK1 Register 1016
4-689. PWRSMWAKEMASK2 Register 1017
4-690. PWRSMMISEVTMASK0 Register 1018
4-691. PWRSMMISEVTMASK1 Register 1019
4-692. PWRSMMISEVTMASK2 Register 1020
4-693. PWRSMWAKESRCSTAT0 Register 1021
4-694. PWRSMWAKESRCSTAT1 Register 1022
4-695. PWRSMWAKESRCSTAT2 Register 1023
4-696. PWRSMEVNTMONSTAT0 Register 1024
4-697. PWRSMEVNTMONSTAT1 Register 1025
4-698. PWRSMEVNTMONSTAT2 Register 1026
4-699. PWRSMWAKESRCSTATCLR0 Register 1027
4-700. PWRSMWAKESRCSTATCLR1 Register 1028
4-701. PWRSMWAKESRCSTATCLR2 Register 1029
4-702. ADCBUFCFG1 Register 1030
4-703. ADCBUFCFG2 Register 1031
4-704. ADCBUFCFG3 Register 1032
4-705. ADCBUFCFG4 Register 1033
4-706. STCPBISTSMCFG1 Register 1034
4-707. STCPBISTSMCFG2 Register 1035
4-708. RTI2EVENTCAPTURESEL Register 1036
4-709. DSSMISC5 Register 1037
4-710. TPTC2WRMPUSTADD0 Register 1040
4-711. TPTC2WRMPUSTADD1 Register 1041
4-712. TPTC2WRMPUSTADD2 Register 1042
4-713. TPTC2WRMPUSTADD3 Register 1043
4-714. TPTC2WRMPUSTADD4 Register 1044
4-715. TPTC2WRMPUSTADD5 Register 1045
4-716. TPTC2WRMPUENDADD0 Register 1046
4-717. TPTC2WRMPUENDADD1 Register 1047
4-718. TPTC2WRMPUENDADD2 Register 1048
4-719. TPTC2WRMPUENDADD3 Register 1049
4-720. TPTC2WRMPUENDADD4 Register 1050
4-721. TPTC2WRMPUENDADD5 Register 1051
4-722. TPTC2WRMPUERRADD Register 1052
4-723. TPTC2RDMPUSTADD0 Register 1053

4-724. TPTC2RDMPUSTADD1 Register 1054
4-725. TPTC2RDMPUSTADD2 Register 1055
4-726. TPTC2RDMPUSTADD3 Register 1056
4-727. TPTC2RDMPUSTADD4 Register 1057
4-728. TPTC2RDMPUSTADD5 Register 1058
4-729. TPTC2RDMPUENDADD0 Register 1059
4-730. TPTC2RDMPUENDADD1 Register 1060
4-731. TPTC2RDMPUENDADD2 Register 1061
4-732. TPTC2RDMPUENDADD3 Register 1062
4-733. TPTC2RDMPUENDADD4 Register 1063
4-734. TPTC2RDMPUENDADD5 Register 1064
4-735. TPTC2RDMPUERRADD Register 1065
4-736. TPTC3WRMPUSTADD0 Register 1066
4-737. TPTC3WRMPUSTADD1 Register 1067
4-738. TPTC3WRMPUSTADD2 Register 1068
4-739. TPTC3WRMPUSTADD3 Register 1069
4-740. TPTC3WRMPUSTADD4 Register 1070
4-741. TPTC3WRMPUSTADD5 Register 1071
4-742. TPTC3WRMPUENDADD0 Register 1072
4-743. TPTC3WRMPUENDADD1 Register 1073
4-744. TPTC3WRMPUENDADD2 Register 1074
4-745. TPTC3WRMPUENDADD3 Register 1075
4-746. TPTC3WRMPUENDADD4 Register 1076
4-747. TPTC3WRMPUENDADD5 Register 1077
4-748. TPTC3WRMPUERRADD Register 1078
4-749. TPTC3RDMPUSTADD0 Register 1079
4-750. TPTC3RDMPUSTADD1 Register 1080
4-751. TPTC3RDMPUSTADD2 Register 1081
4-752. TPTC3RDMPUSTADD3 Register 1082
4-753. TPTC3RDMPUSTADD4 Register 1083
4-754. TPTC3RDMPUSTADD5 Register 1084
4-755. TPTC3RDMPUENDADD0 Register 1085
4-756. TPTC3RDMPUENDADD1 Register 1086
4-757. TPTC3RDMPUENDADD2 Register 1087
4-758. TPTC3RDMPUENDADD3 Register 1088
4-759. TPTC3RDMPUENDADD4 Register 1089
4-760. TPTC3RDMPUENDADD5 Register 1090
4-761. TPTC3RDMPUERRADD Register 1091
4-762. TPTCMPUVALIDCFG2 Register 1092
4-763. TPTCMPUENCFG2 Register 1093
4-764. L3ECCCFG1 Register 1094
4-765. L3ECCCFG2 Register 1095
4-766. DSS2MSSSWIRQ Register 1096
4-767. BSSCTL Register 1099
4-768. DSSCTL Register 1100
4-769. EXTCLKDIV Register 1101
4-770. EXTCLKSRCSEL Register 1102
4-771. EXTCLKCTL Register 1103
4-772. SOFTSYSRST Register 1104

4-773. WDRSTEN Register 1105
4-774. SYSRSTCAUSE Register 1106
4-775. SYSRSTCAUSECLR Register 1107
4-776. MISCCAPT Register 1108
4-777. DCDCCTL0 Register 1109
4-778. DCDCCTL1 Register 1110
4-779. USERMODEEN Register 1111
4-780. LVDSPADCTL0 Register 1112
4-781. LVDSPADCTL1 Register 1113
4-782. DFTREG0 Register 1114
4-783. DFTREG1 Register 1115
4-784. DFTREG5 Register 1116
4-785. MEMINITDONE Register 1117
4-786. SPARE0_3 Register 1118
4-787. MSS_SIGNATURE Register 1119
4-788. GEMBOOTSTCEN Register 1120
4-789. MISCCTL1 Register 1121
4-790. USERMODEEN2 Register 1122
4-791. SYSTICK Register 1123
4-792. SECURECFGREG1 Register 1124
4-793. SECURECFGREG2 Register 1125
4-794. SECURECFGREG3 Register 1126
4-795. SECURECFGREG4 Register 1127
4-796. SECURERAMREG Register 1128
4-797. SPAREMULTIBIT Register 1129
4-798. SPARE4_9 Register 1130
4-799. UID31TO0 Register 1131
4-800. UID63TO32 Register 1132
4-801. UID95TO64 Register 1133
4-802. UID119TO96 Register 1134
4-803. MEMINITSTARTSHMEM Register 1135
4-804. MEMINITDONESHMEM Register 1136
4-805. DSSMEMTAB0 Register 1137
4-806. TCMAMEMTAB Register 1138
4-807. TCMBMEMTAB Register 1139
4-808. SHMEMBANKSEL3TO0 Register 1140
4-809. SHMEMBANKSEL7TO4 Register 1141
4-810. PBISTCLKCTL Register 1142
4-811. SOFTRST1 Register 1145
4-812. SOFTRST2 Register 1146
4-813. CLKDIVCTL0 Register 1147
4-814. CLKSRCSEL0 Register 1148
4-815. CR4CTL Register 1149
4-816. CLKGATE Register 1150
4-817. CLKSRCSEL1 Register 1151
4-818. CURRCLKDIV0 Register 1152
4-819. RTICURRCLKDIV Register 1153
4-820. MEMINITSTART Register 1154
4-821. CURRCLKDIV1 Register 1155

4-822. MEMINITDONE Register 1156
4-823. ECCENMSSGEM Register 1157
4-824. ECCCAPTMSSGEM Register 1158
4-825. ECCENBSSGEM Register 1159
4-826. ECCCAPTBSSGEM Register 1160
4-827. USERMODEEN Register 1161
4-828. NSYSPERUSERMODEN Register 1162
4-829. SECURERAMMMI Register 1163
4-830. SECURERAMECC Register 1164
4-831. ESMGATE0 Register 1165
4-832. ESMGATE1 Register 1166
4-833. ESMGATE2 Register 1167
4-834. ESMGATE3 Register 1168
4-835. ESMGATE4 Register 1169
4-836. KEY Register 1170
4-837. SWIRQA Register 1171
4-838. SWIRQB Register 1172
4-839. MISCCTL0 Register 1173
4-840. ATCMERRCAPTCTL Register 1174
4-841. B0TCMERRCAPTCTL Register 1175
4-842. B1TCMERRCAPTCTL Register 1176
4-843. SOFTCORERST Register 1177
4-844. RSTCAUSE Register 1178
4-845. RSTCAUSECLR Register 1179
4-846. SPITRIGSRC Register 1180
4-847. CLKINUSE Register 1181
4-848. ECCENMSSBSS Register 1182
4-849. ECCCAPTMSSBSS Register 1183
4-850. CLKDIVCTL2 Register 1184
4-851. SWIRQC Register 1185
4-852. GPCFG0 Register 1188
4-853. GPCFG1 Register 1189
4-854. GPCFG2 Register 1190
4-855. GPCFG3 Register 1191
4-856. GPCFG4 Register 1192
4-857. GPCFG6 Register 1193
4-858. GPCFG11 Register 1194
4-859. CCCACFG0 Register 1195
4-860. CCCACFG1 Register 1196
4-861. CCCACFG2 Register 1197
4-862. CCCACFG3 Register 1198
4-863. CCCBCFG0 Register 1199
4-864. CCCBCFG1 Register 1200
4-865. CCCBCFG2 Register 1201
4-866. CCCBCFG3 Register 1202
4-867. CCCACNTVAL Register 1203
4-868. CCCBCNTVAL Register 1204
4-869. CCCABERRSTAT Register 1205
4-870. USERMODEEN Register 1206

4-871. EPWMCFG Register 1207
4-872. DMMSWINT0 Register 1208
4-873. DMMSWINT1 Register 1209
4-874. DMMSWINTSEL0 Register 1210
4-875. DMMSWINTSEL1 Register 1211
4-876. CCCBWDEN Register 1212
4-877. GPIOINTREDGESEL Register 1213
4-878. PWMDMATRIGEN Register 1214
4-879. JTAGTXDATA Register 1215
4-880. JTAGTXCONTROL Register 1216
4-881. JTAGRXDATA Register 1217
4-882. JTAGTXRXACK Register 1218
4-883. JTAGRXCONTROL Register 1219
4-884. MSS2GEMSWIRQ Register 1220
4-885. CSETBFLUSH Register 1221
4-886. RTIEVENTCAPTURESEL Register 1225
4-887. CQCFG1 Register 1226
4-888. TPCCPARSTATCFG Register 1227
4-889. TPTC0WRMPUSTADD0 Register 1228
4-890. TPTC0WRMPUSTADD1 Register 1229
4-891. TPTC0WRMPUSTADD2 Register 1230
4-892. TPTC0WRMPUSTADD3 Register 1231
4-893. TPTC0WRMPUSTADD4 Register 1232
4-894. TPTC0WRMPUSTADD5 Register 1233
4-895. TPTC0WRMPUENDADD0 Register 1234
4-896. TPTC0WRMPUENDADD1 Register 1235
4-897. TPTC0WRMPUENDADD2 Register 1236
4-898. TPTC0WRMPUENDADD3 Register 1237
4-899. TPTC0WRMPUENDADD4 Register 1238
4-900. TPTC0WRMPUENDADD5 Register 1239
4-901. TPTC0WRMPUERRADD Register 1240
4-902. TPTC0RDMPUSTADD0 Register 1241
4-903. TPTC0RDMPUSTADD1 Register 1242
4-904. TPTC0RDMPUSTADD2 Register 1243
4-905. TPTC0RDMPUSTADD3 Register 1244
4-906. TPTC0RDMPUSTADD4 Register 1245
4-907. TPTC0RDMPUSTADD5 Register 1246
4-908. TPTC0RDMPUENDADD0 Register 1247
4-909. TPTC0RDMPUENDADD1 Register 1248
4-910. TPTC0RDMPUENDADD2 Register 1249
4-911. TPTC0RDMPUENDADD3 Register 1250
4-912. TPTC0RDMPUENDADD4 Register 1251
4-913. TPTC0RDMPUENDADD5 Register 1252
4-914. TPTC0RDMPUERRADD Register 1253
4-915. TPTC1WRMPUSTADD0 Register 1254
4-916. TPTC1WRMPUSTADD1 Register 1255
4-917. TPTC1WRMPUSTADD2 Register 1256
4-918. TPTC1WRMPUSTADD3 Register 1257
4-919. TPTC1WRMPUSTADD4 Register 1258

4-920. TPTC1WRMPUSTADD5 Register 1259
4-921. TPTC1WRMPUENDADD0 Register 1260
4-922. TPTC1WRMPUENDADD1 Register 1261
4-923. TPTC1WRMPUENDADD2 Register 1262
4-924. TPTC1WRMPUENDADD3 Register 1263
4-925. TPTC1WRMPUENDADD4 Register 1264
4-926. TPTC1WRMPUENDADD5 Register 1265
4-927. TPTC1WRMPUERRADD Register 1266
4-928. TPTC1RDMPUSTADD0 Register 1267
4-929. TPTC1RDMPUSTADD1 Register 1268
4-930. TPTC1RDMPUSTADD2 Register 1269
4-931. TPTC1RDMPUSTADD3 Register 1270
4-932. TPTC1RDMPUSTADD4 Register 1271
4-933. TPTC1RDMPUSTADD5 Register 1272
4-934. TPTC1RDMPUENDADD0 Register 1273
4-935. TPTC1RDMPUENDADD1 Register 1274
4-936. TPTC1RDMPUENDADD2 Register 1275
4-937. TPTC1RDMPUENDADD3 Register 1276
4-938. TPTC1RDMPUENDADD4 Register 1277
4-939. TPTC1RDMPUENDADD5 Register 1278
4-940. TPTC1RDMPUERRADD Register 1279
4-941. TPTCMPUVALIDCFG Register 1280
4-942. TPTCMPUENCFG Register 1281
4-943. TESTPATTERNRX1ICFG Register 1282
4-944. TESTPATTERNRX2ICFG Register 1283
4-945. TESTPATTERNRX3ICFG Register 1284
4-946. TESTPATTERNRX4ICFG Register 1285
4-947. TESTPATTERNRX1QCFG Register 1286
4-948. TESTPATTERNRX2QCFG Register 1287
4-949. TESTPATTERNRX3QCFG Register 1288
4-950. TESTPATTERNRX4QCFG Register 1289
4-951. TESTPATTERNVLDCFG Register 1290
4-952. TPCC1PARSTATCFG Register 1291
4-953. DMMSWINT1 Register 1292
4-954. DSSINTRCFG Register 1293
4-955. MPUMSTIDCFG1 Register 1294
4-956. MPUMSTIDCFG2 Register 1295
4-957. MPUMSTIDCFG3 Register 1296
4-958. HSRAM1ECCCFG Register 1297
4-959. DATATRRAMECCCFG Register 1298
4-960. ADCBUFPINGECCCFG Register 1299
4-961. ADCBUFPONGECCCFG Register 1300
4-962. UMAP0PARITYCFG1 Register 1301
4-963. UMAP0PARITYCFG2 Register 1302
4-964. UMAP0PARITYCFG3 Register 1303
4-965. UMAP1PARITYCFG1 Register 1304
4-966. UMAP1PARITYCFG2 Register 1305
4-967. UMAP1PARITYCFG3 Register 1306
4-968. ESMGRP2MASKCFG Register 1307

4-969. L2MEMINITCFG1 Register 1308
4-970. L2MEMINITCFG2 Register 1310
4-971. GEMRSTCAUSE Register 1311
4-972. GEMPWRSMCFG4 Register 1312
4-973. PWRSMWAKEMASK0 Register 1313
4-974. PWRSMWAKEMASK1 Register 1314
4-975. PWRSMWAKEMASK2 Register 1315
4-976. PWRSMMISEVTMASK0 Register 1316
4-977. PWRSMMISEVTMASK1 Register 1317
4-978. PWRSMMISEVTMASK2 Register 1318
4-979. PWRSMWAKESRCSTAT0 Register 1319
4-980. PWRSMWAKESRCSTAT1 Register 1320
4-981. PWRSMWAKESRCSTAT2 Register 1321
4-982. PWRSMEVNTMONSTAT0 Register 1322
4-983. PWRSMEVNTMONSTAT1 Register 1323
4-984. PWRSMEVNTMONSTAT2 Register 1324
4-985. PWRSMWAKESRCSTATCLR0 Register 1325
4-986. PWRSMWAKESRCSTATCLR1 Register 1326
4-987. PWRSMWAKESRCSTATCLR2 Register 1327
4-988. ADCBUFCFG1 Register 1328
4-989. ADCBUFCFG2 Register 1329
4-990. ADCBUFCFG3 Register 1330
4-991. ADCBUFCFG4 Register 1331
4-992. STCPBISTSMCFG1 Register 1332
4-993. STCPBISTSMCFG2 Register 1333
4-994. RTI2EVENTCAPTURESEL Register 1334
4-995. DSSMISC5 Register 1335
4-996. TPTC2WRMPUSTADD0 Register 1338
4-997. TPTC2WRMPUSTADD1 Register 1339
4-998. TPTC2WRMPUSTADD2 Register 1340
4-999. TPTC2WRMPUSTADD3 Register 1341
4-1000. TPTC2WRMPUSTADD4 Register 1342
4-1001. TPTC2WRMPUSTADD5 Register 1343
4-1002. TPTC2WRMPUENDADD0 Register 1344
4-1003. TPTC2WRMPUENDADD1 Register 1345
4-1004. TPTC2WRMPUENDADD2 Register 1346
4-1005. TPTC2WRMPUENDADD3 Register 1347
4-1006. TPTC2WRMPUENDADD4 Register 1348
4-1007. TPTC2WRMPUENDADD5 Register 1349
4-1008. TPTC2WRMPUERRADD Register 1350
4-1009. TPTC2RDMPUSTADD0 Register 1351
4-1010. TPTC2RDMPUSTADD1 Register 1352
4-1011. TPTC2RDMPUSTADD2 Register 1353
4-1012. TPTC2RDMPUSTADD3 Register 1354
4-1013. TPTC2RDMPUSTADD4 Register 1355
4-1014. TPTC2RDMPUSTADD5 Register 1356
4-1015. TPTC2RDMPUENDADD0 Register 1357
4-1016. TPTC2RDMPUENDADD1 Register 1358
4-1017. TPTC2RDMPUENDADD2 Register 1359

4-1018. TPTC2RDMPUENDADD3 Register 1360
4-1019. TPTC2RDMPUENDADD4 Register 1361
4-1020. TPTC2RDMPUENDADD5 Register 1362
4-1021. TPTC2RDMPUERRADD Register 1363
4-1022. TPTC3WRMPUSTADD0 Register 1364
4-1023. TPTC3WRMPUSTADD1 Register 1365
4-1024. TPTC3WRMPUSTADD2 Register 1366
4-1025. TPTC3WRMPUSTADD3 Register 1367
4-1026. TPTC3WRMPUSTADD4 Register 1368
4-1027. TPTC3WRMPUSTADD5 Register 1369
4-1028. TPTC3WRMPUENDADD0 Register 1370
4-1029. TPTC3WRMPUENDADD1 Register 1371
4-1030. TPTC3WRMPUENDADD2 Register 1372
4-1031. TPTC3WRMPUENDADD3 Register 1373
4-1032. TPTC3WRMPUENDADD4 Register 1374
4-1033. TPTC3WRMPUENDADD5 Register 1375
4-1034. TPTC3WRMPUERRADD Register 1376
4-1035. TPTC3RDMPUSTADD0 Register 1377
4-1036. TPTC3RDMPUSTADD1 Register 1378
4-1037. TPTC3RDMPUSTADD2 Register 1379
4-1038. TPTC3RDMPUSTADD3 Register 1380
4-1039. TPTC3RDMPUSTADD4 Register 1381
4-1040. TPTC3RDMPUSTADD5 Register 1382
4-1041. TPTC3RDMPUENDADD0 Register 1383
4-1042. TPTC3RDMPUENDADD1 Register 1384
4-1043. TPTC3RDMPUENDADD2 Register 1385
4-1044. TPTC3RDMPUENDADD3 Register 1386
4-1045. TPTC3RDMPUENDADD4 Register 1387
4-1046. TPTC3RDMPUENDADD5 Register 1388
4-1047. TPTC3RDMPUERRADD Register 1389
4-1048. TPTCMPUVALIDCFG2 Register 1390
4-1049. TPTCMPUENCFG2 Register 1391
4-1050. L3ECCCFG1 Register 1392
4-1051. L3ECCCFG2 Register 1393
4-1052. DSS2MSSSWIRQ Register 1394
5-1. RADAR Subsystem Block Diagram 1396
5-2. Programming Model 1397
5-3. Clocking Subsystem 1398
5-4. Transmit Subsystem 1399
5-5. Receive Subsystem (Per Channel) 1399
6-1. Cortex-R4F Integration 1401
7-1. DMA Block Diagram 1403
7-2. Example of a DMA Transfer Using Frame Trigger Source 1404
7-3. Example of a DMA Transfer Using Block Trigger Source 1404
7-4. DMA Request Mapping and Control Packet Organization 1406
7-5. Control Packet Organization and Memory Map 1406
7-6. DMA Transfer Example 1 1408
7-7. DMA Indexing Example 1 1408
7-8. DMA Indexing Example 2 1409

7-9. Fixed Priority Scheme 1409
7-10. Example of Priority Queues 1410
7-11. Example Channel Assignments 1411
7-12. Example of DMA Data Unpacking 1412
7-13. Example of DMA Data Packing 1413
7-14. DMA Interrupts 1416
7-15. Detailed Interrupt Structure (Frame Transfer Complete Path) 1416
7-16. Example of Channel Chaining 1419
7-17. Example of Protection Mechanism 1420
7-18. GCTRL Register 1425
7-19. PEND Register 1426
7-20. FBREG Register 1427
7-21. DMASTAT Register 1428
7-22. HWCHENAS Register 1429
7-23. HWCHENAR Register 1430
7-24. SWCHENAS Register 1431
7-25. SWCHENAR Register 1432
7-26. CHPRIOS Register 1433
7-27. CHPRIOR Register 1434
7-28. GCHIENAS Register 1435
7-29. GCHIENAR Register 1436
7-30. DREQASI0 Register 1437
7-31. DREQASI1 Register 1438
7-32. DREQASI2 Register 1439
7-33. DREQASI3 Register 1440
7-34. DREQASI4 Register 1441
7-35. DREQASI5 Register 1442
7-36. DREQASI6 Register 1443
7-37. DREQASI7 Register 1444
7-38. PAR0 Register 1445
7-39. PAR1 Register 1446
7-40. PAR2 Register 1447
7-41. PAR3 Register 1448
7-42. FTCMAP Register 1449
7-43. LFSMAP Register 1450
7-44. HBCMAP Register 1451
7-45. BTCMAP Register 1452
7-46. BERMAP Register 1453
7-47. FTCINTENAS Register 1454
7-48. FTCINTENAR Register 1455
7-49. LFSINTENAS Register 1456
7-50. LFSINTENAR Register 1457
7-51. HBCINTENAS Register 1458
7-52. HBCINTENAR Register 1459
7-53. BTCINTENAS Register 1460
7-54. BTCINTENAR Register 1461
7-55. GINTFLAG Register 1462
7-56. FTCFLAG Register 1463
7-57. LFSFLAG Register 1464

7-58. HBCFLAG Register 1465
7-59. BTCFLAG Register 1466
7-60. BERFLAG Register 1467
7-61. FTCAOFFSET Register 1468
7-62. LFSAOFFSET Register 1469
7-63. HBCAOFFSET Register 1470
7-64. BTCAOFFSET Register 1471
7-65. BERAOFFSET Register 1472
7-66. FTCBOFFSET Register 1473
7-67. LFSBOFFSET Register 1474
7-68. HBCBOFFSET Register 1475
7-69. BTCBOFFSET Register 1476
7-70. BERBOFFSET Register 1477
7-71. PTCRL Register 1478
7-72. RTCTRL Register 1480
7-73. DCTRL Register 1481
7-74. WPR Register 1482
7-75. WMR Register 1483
7-76. PAACSADDR Register 1484
7-77. PAACDADDR Register 1485
7-78. PAACTC Register 1486
7-79. PBACSADDR Register 1487
7-80. PBACDADDR Register 1488
7-81. PBACTC Register 1489
7-82. DMAPCR Register 1490
7-83. DMAPAR Register 1491
7-84. DMAMPCTRL Register 1492
7-85. DMAMPST Register 1494
7-86. DMAMPR0S Register 1495
7-87. DMAMPR0E Register 1496
7-88. DMAMPR1S Register 1497
7-89. DMAMPR1E Register 1498
7-90. DMAMPR2S Register 1499
7-91. DMAMPR2E Register 1500
7-92. DMAMPR3S Register 1501
7-93. DMAMPR3E Register 1502
7-94. Initial Source Address (ISADDR) [offset = 00] 1503
7-95. Initial Destination Address Register (IDADDR) [offset = 04h] 1503
7-96. Initial Transfer Count Register (ITCOUNT) [offset = 08h] 1504
7-97. Channel Control Register (CHCTRL) [offset = 10h] 1504
7-98. Element Index Offset Register (EIOFF) [offset = 14h] 1506
7-99. Frame Index Offset Register (FIOFF) [offset = 18h] 1506
7-100. Current Source Address Register (CSADDR) [offset = 800h] 1507
7-101. Current Destination Address Register (CDADDR) [offset = 804h] 1507
7-102. Current Transfer Count Register (CTCOUNT) [offset = 808h] 1508
8-1. Device Level Interrupt Block Diagram 1511
8-2. VIM Interrupt Handling Block Diagram 1514
8-3. VIM Channel Mapping 1515
8-4. VIM in Default State 1516

8-5. VIM in a Programmed State 1516
8-6. Interrupt Channel Management 1517
8-7. VIM Interrupt Address Memory Map 1518
8-8. ECC Bits Mapping 1522
8-9. Detail of the IRQ Input 1523
8-10. Capture Event Sources 1524
8-11. Interrupt Vector Table ECC Status Register (ECCSTAT) [offset = ECh] 1528
8-12. Interrupt Vector Table ECC Control Register (ECCCTL) [offset = F0h] 1529
8-13. Uncorrectable Error Address Register (UERRADDR) [offset = F4h] 1530
8-14. Fallback Vector Address Register (FBVECADDR) [offset = F8h] 1530
8-15. Single Bit Error Address Register (SBERRADDR) [offset = FCh] 1531
8-16. IRQ Index Offset Vector Register (IRQINDEX) [offset = 00h] 1532
8-17. FIQ Index Offset Vector Register (FIQINDEX) [offset = F04h] 1532
8-18. FIQ/IRQ Program Control Register 0 (FIRQPR0) [offset = 10h] 1533
8-19. FIQ/IRQ Program Control Register 1 (FIRQPR1) [offset = F14h] 1533
8-20. FIQ/IRQ Program Control Register 2 (FIRQPR2) [offset = 18h] 1533
8-21. FIQ/IRQ Program Control Register 3 (FIRQPR3) [offset = 1Ch] 1533
8-22. Pending Interrupt Read Location Register 0 (INTREQ0) [offset = 20h] 1534
8-23. Pending Interrupt Read Location Register 1 (INTREQ1) [offset = 24h] 1534
8-24. Pending Interrupt Read Location Register 2 (INTREQ2) [offset = 28h] 1534
8-25. Pending Interrupt Read Location Register 3 (INTREQ3) [offset = 2Ch] 1534
8-26. Interrupt Enable Set Register 0 (REQENASET0) [offset = 30h] 1535
8-27. Interrupt Enable Set Register 1 (REQENASET1) [offset = 34h] 1535
8-28. Interrupt Enable Set Register 2 (REQENASET2) [offset = 38h] 1535
8-29. Interrupt Enable Set Register 3 (REQENASET3) [offset = 3Ch] 1535
8-30. Interrupt Enable Clear Register 0 (REQENACLR0) [offset = 40h] 1536
8-31. Interrupt Enable Clear Register 1 (REQENACLR1) [offset = 44h] 1536
8-32. Interrupt Enable Clear Register 2 (REQENACLR2) [offset = 48h] 1536
8-33. Interrupt Enable Clear Register 3 (REQENACLR3) [offset = 4Ch] 1536
8-34. Wake-Up Enable Set Register 0 (WAKEENASET0) [offset = 50h] 1537
8-35. Wake-Up Enable Set Register 1 (WAKEENASET1) [offset = 54h] 1537
8-36. Wake-Up Enable Set Register 2 (WAKEENASET2) [offset = 58h] 1537
8-37. Wake-Up Enable Set Register 3 (WAKEENASET3) [offset = 5Ch] 1537
8-38. Wake-Up Enable Clear Register 0 (WAKEENACLR0) [offset = 60h] 1538
8-39. Wake-Up Enable Clear Register 1 (WAKEENACLR1) [offset = 64h] 1538
8-40. Wake-Up Enable Clear Register 2 (WAKEENACLR2) [offset = 68h] 1538
8-41. Wake-Up Enable Clear Register 3 (WAKEENACLR3) [offset = 6Ch] 1538
8-42. IRQ Interrupt Vector Register (IRQVECREG) [offset = 70h] 1539
8-43. IRQ Interrupt Vector Register (*FIQVECREG*) [offset = 74h] 1539
8-44. Capture Event Register (CAPEVT) [offset = 78h] 1540
8-45. Interrupt Control Registers (CHANCTRL[0:31]) [offset = 80h-FCh] 1541
9-1. TMS320C674x Megamodule Block Diagram 1544
10-1. DSS_L3 Memory Organization 1549
10-2. DSS_L3 Memory Organization 1551
10-3. DSS_L3 Memory Organization 1553
12-1. EDMA Module Overview 1557
12-2. EDMA Controller Block Diagram 1559
12-3. EDMA Channel Controller Block Diagram 1560
12-4. TPTC Block Diagram 1561

12-5. Definition of ACNT, BCNT, and CCNT 1563
12-6. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3) 1563
12-7. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3) 1564
12-8. PaRAM Set 1566
12-9. Linked Transfer 1574
12-10. Link-to-Self Transfer 1575
12-11. DMA Channel and QDMA Channel to PaRAM Mapping 1581
12-12. QDMA Channel to PaRAM Mapping 1582
12-13. Shadow Region Registers 1583
12-14. Interrupt Diagram 1589
12-15. Error Interrupt Operation 1593
12-16. PaRAM Set Content for Proxy Memory Protection Example 1598
12-17. Channel Options Parameter (OPT) Example 1598
12-18. Proxy Memory Protection Example 1599
12-19. EDMA Prioritization 1605
12-20. Block Move Example 1607
12-21. Block Move Example PaRAM Configuration 1608
12-22. Subframe Extraction Transfer 1609
12-23. Subframe Extraction Example PaRAM Configuration 1609
12-24. Data Sorting Example 1610
12-25. Data Sorting Example PaRAM Configuration 1611
12-26. EDMA_TPCC_PID Register 1620
12-27. EDMA_TPCC_CCCFG Register 1621
12-28. EDMA_TPCC_QCHMAPN Register 1623
12-29. EDMA_TPCC_DMAQNUMN Register 1624
12-30. EDMA_TPCC_QDMAQNUM Register 1625
12-31. EDMA_TPCC_QUETCMAP Register 1626
12-32. EDMA_TPCC_QUEPRI Register 1627
12-33. EDMA_TPCC_EMR Register 1628
12-34. EDMA_TPCC_EMRH Register 1630
12-35. EDMA_TPCC_EMCR Register 1632
12-36. EDMA_TPCC_EMCRH Register 1634
12-37. EDMA_TPCC_QEMR Register 1636
12-38. EDMA_TPCC_QEMCR Register 1637
12-39. EDMA_TPCC_CCERR Register 1638
12-40. EDMA_TPCC_CCERRCLR Register 1640
12-41. EDMA_TPCC_EEVAL Register 1641
12-42. EDMA_TPCC_DRAEN Register 1642
12-43. EDMA_TPCC_DRAEM Register 1643
12-44. EDMA_TPCC_DRAEHM Register 1645
12-45. EDMA_TPCC_QRAEN Register 1647
12-46. EDMA_TPCC_QNEM Register 1648
12-47. EDMA_TPCC_QNE0 Register 1649
12-48. EDMA_TPCC_QNE1 Register 1650
12-49. EDMA_TPCC_QNE2 Register 1651
12-50. EDMA_TPCC_QNE3 Register 1652
12-51. EDMA_TPCC_QNE4 Register 1653
12-52. EDMA_TPCC_QNE5 Register 1654
12-53. EDMA_TPCC_QNE6 Register 1655

12-54. EDMA_TPCC_QNE7 Register 1656
12-55. EDMA_TPCC_QNE8 Register 1657
12-56. EDMA_TPCC_QNE9 Register 1658
12-57. EDMA_TPCC_QNE10 Register 1659
12-58. EDMA_TPCC_QNE11 Register 1660
12-59. EDMA_TPCC_QNE12 Register 1661
12-60. EDMA_TPCC_QNE13 Register 1662
12-61. EDMA_TPCC_QNE14 Register 1663
12-62. EDMA_TPCC_QNE15 Register 1664
12-63. EDMA_TPCC_QSTATN Register 1665
12-64. EDMA_TPCC_QWMTHRA Register 1666
12-65. EDMA_TPCC_CCSTAT Register 1667
12-66. EDMA_TPCC_AETCTL Register 1669
12-67. EDMA_TPCC_AETSTAT Register 1670
12-68. EDMA_TPCC_AETCMD Register 1671
12-69. EDMA_TPCC_ER Register 1672
12-70. EDMA_TPCC_ERH Register 1674
12-71. EDMA_TPCC_ECR Register 1676
12-72. EDMA_TPCC_ECRH Register 1678
12-73. EDMA_TPCC_ESR Register 1680
12-74. EDMA_TPCC_ESRH Register 1682
12-75. EDMA_TPCC_CER Register 1684
12-76. EDMA_TPCC_CERH Register 1686
12-77. EDMA_TPCC_EER Register 1688
12-78. EDMA_TPCC_EERH Register 1690
12-79. EDMA_TPCC_EECR Register 1692
12-80. EDMA_TPCC_EECRH Register 1694
12-81. EDMA_TPCC_EESR Register 1696
12-82. EDMA_TPCC_EESRH Register 1698
12-83. EDMA_TPCC_SER Register 1700
12-84. EDMA_TPCC_SERH Register 1702
12-85. EDMA_TPCC_SECR Register 1704
12-86. EDMA_TPCC_SECRH Register 1706
12-87. EDMA_TPCC_IER Register 1708
12-88. EDMA_TPCC_IERH Register 1710
12-89. EDMA_TPCC_IECR Register 1712
12-90. EDMA_TPCC_IECRH Register 1714
12-91. EDMA_TPCC_IESR Register 1716
12-92. EDMA_TPCC_IESRH Register 1718
12-93. EDMA_TPCC_IPR Register 1720
12-94. EDMA_TPCC_IPRH Register 1722
12-95. EDMA_TPCC_ICR Register 1724
12-96. EDMA_TPCC_ICRH Register 1726
12-97. EDMA_TPCC_IEVAL Register 1728
12-98. EDMA_TPCC_QER Register 1729
12-99. EDMA_TPCC_QEER Register 1730
12-100. EDMA_TPCC_QEECR Register 1731
12-101. EDMA_TPCC_QEESR Register 1732
12-102. EDMA_TPCC_QSER Register 1733

12-103. EDMA_TPCC_QSECR Register 1734
12-104. EDMA_TPCC_SHADOW_N Register 1735
12-105. EDMA_TPCC_ER_RN Register 1736
12-106. EDMA_TPCC_ERH_RN Register 1738
12-107. EDMA_TPCC_ECR_RN Register 1740
12-108. EDMA_TPCC_ECRH_RN Register 1742
12-109. EDMA_TPCC_ESR_RN Register 1744
12-110. EDMA_TPCC_ESRH_RN Register 1746
12-111. EDMA_TPCC_CER_RN Register 1748
12-112. EDMA_TPCC_CERH_RN Register 1750
12-113. EDMA_TPCC_EER_RN Register 1752
12-114. EDMA_TPCC_EERH_RN Register 1754
12-115. EDMA_TPCC_EECR_RN Register 1756
12-116. EDMA_TPCC_EECRH_RN Register 1758
12-117. EDMA_TPCC_EESR_RN Register 1760
12-118. EDMA_TPCC_EESRH_RN Register 1762
12-119. EDMA_TPCC_SER_RN Register 1764
12-120. EDMA_TPCC_SERH_RN Register 1766
12-121. EDMA_TPCC_SECR_RN Register 1768
12-122. EDMA_TPCC_SECRH_RN Register 1770
12-123. EDMA_TPCC_IER_RN Register 1772
12-124. EDMA_TPCC_IERH_RN Register 1774
12-125. EDMA_TPCC_IECR_RN Register 1776
12-126. EDMA_TPCC_IECRH_RN Register 1778
12-127. EDMA_TPCC_IESR_RN Register 1780
12-128. EDMA_TPCC_IESRH_RN Register 1782
12-129. EDMA_TPCC_IPR_RN Register 1784
12-130. EDMA_TPCC_IPRH_RN Register 1786
12-131. EDMA_TPCC_ICR_RN Register 1788
12-132. EDMA_TPCC_ICRH_RN Register 1790
12-133. EDMA_TPCC_IEVAL_RN Register 1792
12-134. EDMA_TPCC_QER_RN Register 1793
12-135. EDMA_TPCC_QEER_RN Register 1794
12-136. EDMA_TPCC_QEECR_RN Register 1795
12-137. EDMA_TPCC_QEESR_RN Register 1796
12-138. EDMA_TPCC_QSER_RN Register 1797
12-139. EDMA_TPCC_QSECR_RN Register 1798
12-140. EDMA_TPCC_PARAMSET Register 1799
12-141. EDMA_TPCC_OPT Register 1800
12-142. EDMA_TPCC_SRC Register 1802
12-143. EDMA_TPCC_ABCNT Register 1803
12-144. EDMA_TPCC_DST Register 1804
12-145. EDMA_TPCC_BIDX Register 1805
12-146. EDMA_TPCC_LNK Register 1806
12-147. EDMA_TPCC_CIDX Register 1807
12-148. EDMA_TPCC_CCNT Register 1808
12-149. EDMA_TPTC_PID Register 1810
12-150. EDMA_TPTC_TCCFG Register 1811
12-151. EDMA_TPTC_TCSTAT Register 1812

12-152. EDMA_TPTC_INTSTAT Register 1813
12-153. EDMA_TPTC_INTEN Register 1814
12-154. EDMA_TPTC_INTCLR Register 1815
12-155. EDMA_TPTC_INTCMD Register 1816
12-156. EDMA_TPTC_ERRSTAT Register 1817
12-157. EDMA_TPTC_ERREN Register 1818
12-158. EDMA_TPTC_ERRCLR Register 1819
12-159. EDMA_TPTC_ERRDET Register 1820
12-160. EDMA_TPTC_ERRCMD Register 1821
12-161. EDMA_TPTC_RDRATE Register 1822
12-162. EDMA_TPTC_POPT Register 1823
12-163. EDMA_TPTC_PSRC Register 1824
12-164. EDMA_TPTC_PCNT Register 1825
12-165. EDMA_TPTC_PDST Register 1826
12-166. EDMA_TPTC_PBIDX Register 1827
12-167. EDMA_TPTC_PMPPRXY Register 1828
12-168. EDMA_TPTC_SAOPT Register 1829
12-169. EDMA_TPTC_SASRC Register 1830
12-170. EDMA_TPTC_SACNT Register 1831
12-171. EDMA_TPTC_SABIDX Register 1832
12-172. EDMA_TPTC_SAMPPRXY Register 1833
12-173. EDMA_TPTC_SACNTRLD Register 1834
12-174. EDMA_TPTC_SASRCBREF Register 1835
12-175. EDMA_TPTC_SADSTBREF Register 1836
12-176. EDMA_TPTC_DFCNTRLD Register 1837
12-177. EDMA_TPTC_DFSRCBREF Register 1838
12-178. EDMA_TPTC_DFOPT Register 1839
12-179. EDMA_TPTC_DFSRC Register 1840
12-180. EDMA_TPTC_DFCNT Register 1841
12-181. EDMA_TPTC_DFDST Register 1842
12-182. EDMA_TPTC_DFBIDX Register 1843
12-183. EDMA_TPTC_DFMPPRXY Register 1844
13-1. ADC Buffer Block Diagram for the 14xx 1846
13-2. ADC Buffer Block Diagram for the 16xx/18xx/68xx 1846
13-3. Single-Chirp Mode 1847
13-4. Multi-Chirp Mode 1848
14-1. HSI Overview 1852
14-2. CBUFF Sequencing 1854
14-3. CBUFF Linklist Concept 1855
14-4. Default Use Case 1857
14-5. LVDS Interface Timings 1859
14-6. CONFIG_REG_0 Register 1866
14-7. CFG_SPHDR_ADDRESS Register 1868
14-8. CFG_CMD_HSVAL Register 1869
14-9. CFG_CMD_HEVAL Register 1870
14-10. CFG_CMD_VSVAL Register 1871
14-11. CFG_CMD_VEVAL Register 1872
14-12. CFG_LPHDR_ADDRESS Register 1873
14-13. CFG_CHIRPS_PER_FRAME Register 1874

14-14. CFG_FIFO_FREE_THRESHOLD Register 1875
14-15. CFG_LPPYLD_ADDRESS Register 1876
14-16. CFG_DATA_LL0 Register 1877
14-17. CFG_DATA_LL0_LPHDR_VAL Register 1878
14-18. CFG_DATA_LL0_THRESHOLD Register 1879
14-19. CFG_DATA_LL1 Register 1880
14-20. CFG_DATA_LL1_LPHDR_VAL Register 1881
14-21. CFG_DATA_LL1_THRESHOLD Register 1882
14-22. CFG_DATA_LL2 Register 1883
14-23. CFG_DATA_LL2_LPHDR_VAL Register 1884
14-24. CFG_DATA_LL2_THRESHOLD Register 1885
14-25. CFG_DATA_LL3 Register 1886
14-26. CFG_DATA_LL3_LPHDR_VAL Register 1887
14-27. CFG_DATA_LL3_THRESHOLD Register 1888
14-28. CFG_DATA_LL4 Register 1889
14-29. CFG_DATA_LL4_LPHDR_VAL Register 1890
14-30. CFG_DATA_LL4_THRESHOLD Register 1891
14-31. CFG_DATA_LL5 Register 1892
14-32. CFG_DATA_LL5_LPHDR_VAL Register 1893
14-33. CFG_DATA_LL5_THRESHOLD Register 1894
14-34. CFG_DATA_LL6 Register 1895
14-35. CFG_DATA_LL6_LPHDR_VAL Register 1896
14-36. CFG_DATA_LL6_THRESHOLD Register 1897
14-37. CFG_DATA_LL7 Register 1898
14-38. CFG_DATA_LL7_LPHDR_VAL Register 1899
14-39. CFG_DATA_LL7_THRESHOLD Register 1900
14-40. CFG_DATA_LL8 Register 1901
14-41. CFG_DATA_LL8_LPHDR_VAL Register 1902
14-42. CFG_DATA_LL8_THRESHOLD Register 1903
14-43. CFG_DATA_LL9 Register 1904
14-44. CFG_DATA_LL9_LPHDR_VAL Register 1905
14-45. CFG_DATA_LL9_THRESHOLD Register 1906
14-46. CFG_DATA_LL10 Register 1907
14-47. CFG_DATA_LL10_LPHDR_VAL Register 1908
14-48. CFG_DATA_LL10_THRESHOLD Register 1909
14-49. CFG_DATA_LL11 Register 1910
14-50. CFG_DATA_LL11_LPHDR_VAL Register 1911
14-51. CFG_DATA_LL11_THRESHOLD Register 1912
14-52. CFG_DATA_LL12 Register 1913
14-53. CFG_DATA_LL12_LPHDR_VAL Register 1914
14-54. CFG_DATA_LL12_THRESHOLD Register 1915
14-55. CFG_DATA_LL13 Register 1916
14-56. CFG_DATA_LL13_LPHDR_VAL Register 1917
14-57. CFG_DATA_LL13_THRESHOLD Register 1918
14-58. CFG_DATA_LL14 Register 1919
14-59. CFG_DATA_LL14_LPHDR_VAL Register 1920
14-60. CFG_DATA_LL14_THRESHOLD Register 1921
14-61. CFG_DATA_LL15 Register 1922
14-62. CFG_DATA_LL15_LPHDR_VAL Register 1923

14-63. CFG_DATA_LL15_THRESHOLD Register 1924
14-64. CFG_DATA_LL16 Register 1925
14-65. CFG_DATA_LL16_LPHDR_VAL Register 1926
14-66. CFG_DATA_LL16_THRESHOLD Register 1927
14-67. CFG_DATA_LL17 Register 1928
14-68. CFG_DATA_LL17_LPHDR_VAL Register 1929
14-69. CFG_DATA_LL17_THRESHOLD Register 1930
14-70. CFG_DATA_LL18 Register 1931
14-71. CFG_DATA_LL18_LPHDR_VAL Register 1932
14-72. CFG_DATA_LL18_THRESHOLD Register 1933
14-73. CFG_DATA_LL19 Register 1934
14-74. CFG_DATA_LL19_LPHDR_VAL Register 1935
14-75. CFG_DATA_LL19_THRESHOLD Register 1936
14-76. CFG_DATA_LL20 Register 1937
14-77. CFG_DATA_LL20_LPHDR_VAL Register 1938
14-78. CFG_DATA_LL20_THRESHOLD Register 1939
14-79. CFG_DATA_LL21 Register 1940
14-80. CFG_DATA_LL21_LPHDR_VAL Register 1941
14-81. CFG_DATA_LL21_THRESHOLD Register 1942
14-82. CFG_DATA_LL22 Register 1943
14-83. CFG_DATA_LL22_LPHDR_VAL Register 1944
14-84. CFG_DATA_LL22_THRESHOLD Register 1945
14-85. CFG_DATA_LL23 Register 1946
14-86. CFG_DATA_LL23_LPHDR_VAL Register 1947
14-87. CFG_DATA_LL23_THRESHOLD Register 1948
14-88. CFG_DATA_LL24 Register 1949
14-89. CFG_DATA_LL24_LPHDR_VAL Register 1950
14-90. CFG_DATA_LL24_THRESHOLD Register 1951
14-91. CFG_DATA_LL25 Register 1952
14-92. CFG_DATA_LL25_LPHDR_VAL Register 1953
14-93. CFG_DATA_LL25_THRESHOLD Register 1954
14-94. CFG_DATA_LL26 Register 1955
14-95. CFG_DATA_LL26_LPHDR_VAL Register 1956
14-96. CFG_DATA_LL26_THRESHOLD Register 1957
14-97. CFG_DATA_LL27 Register 1958
14-98. CFG_DATA_LL27_LPHDR_VAL Register 1959
14-99. CFG_DATA_LL27_THRESHOLD Register 1960
14-100. CFG_DATA_LL28 Register 1961
14-101. CFG_DATA_LL28_LPHDR_VAL Register 1962
14-102. CFG_DATA_LL28_THRESHOLD Register 1963
14-103. CFG_DATA_LL29 Register 1964
14-104. CFG_DATA_LL29_LPHDR_VAL Register 1965
14-105. CFG_DATA_LL29_THRESHOLD Register 1966
14-106. CFG_DATA_LL30 Register 1967
14-107. CFG_DATA_LL30_LPHDR_VAL Register 1968
14-108. CFG_DATA_LL30_THRESHOLD Register 1969
14-109. CFG_DATA_LL31 Register 1970
14-110. CFG_DATA_LL31_LPHDR_VAL Register 1971
14-111. CFG_DATA_LL31_THRESHOLD Register 1972

14-112. CFG_LVDS_MAPPING_LANE0_FMT_0 Register 1973
14-113. CFG_LVDS_MAPPING_LANE1_FMT_0 Register 1974
14-114. CFG_LVDS_MAPPING_LANE2_FMT_0 Register 1975
14-115. CFG_LVDS_MAPPING_LANE3_FMT_0 Register 1976
14-116. CFG_LVDS_MAPPING_LANE0_FMT_1 Register 1977
14-117. CFG_LVDS_MAPPING_LANE1_FMT_1 Register 1978
14-118. CFG_LVDS_MAPPING_LANE2_FMT_1 Register 1979
14-119. CFG_LVDS_MAPPING_LANE3_FMT_1 Register 1980
14-120. CFG_LVDS_GEN_0 Register 1981
14-121. CFG_LVDS_GEN_1 Register 1982
14-122. CFG_LVDS_GEN_2 Register 1983
14-123. CFG_MASK_REG0 Register 1984
14-124. CFG_MASK_REG1 Register 1985
14-125. CFG_MASK_REG2 Register 1986
14-126. CFG_MASK_REG3 Register 1987
14-127. STAT_CBUFF_REG0 Register 1988
14-128. STAT_CBUFF_REG1 Register 1989
14-129. STAT_CBUFF_REG2 Register 1990
14-130. STAT_CBUFF_REG3 Register 1991
14-131. CLR_CBUFF_REG0 Register 1992
14-132. STAT_CBUFF_ECC_REG Register 1993
14-133. MASK_CBUFF_ECC_REG Register 1994
14-134. CLR_CBUFF_ECC_REG Register 1995
14-135. STAT_SAFETY Register 1996
14-136. MASK_SAFETY Register 1997
14-137. CLR_SAFETY Register 1998
14-138. CSI2 Transmitter/Receiver Data Flow 1999
14-139. Low Level Protocol Packet Overview 2001
14-140. Long Packet Structure 2002
14-141. Short Packet Structure 2003
14-142. Data Identifier Byte 2003
14-143. Logical Channel Block Diagram (Receiver) 2004
14-144. Interleaved Video Data Streams Examples 2004
14-145. Four-Data Lane Configuration 2006
14-146. Three-Data Lane Configuration 2007
14-147. Two-Data Lane Configuration 2008
14-148. One-Data Lane Configuration 2008
14-149. CSI2 Protocol Engine 2009
14-150. Interface Data Configuration 2010
14-151. LP to HS Timing 2011
14-152. HS to LP Timing 2013
14-153. 64-Bit ECC Generation on TX Side 2015
14-154. Checksum Transmission 2016
14-155. 16 Bit CRC Generation Using a Shift Register 2016
14-156. Complex I/O Power FSM 2017
14-157. HS Clock Transmission 2019
14-158. HS Data Transmission 2020
14-159. CSI2_REVISION Register 2031
14-160. CSI2_SYSCONFIG Register 2032

14-161. CSI2_SYSSTATUS Register 2033
14-162. CSI2_IRQSTATUS Register 2034
14-163. CSI2_IRQENABLE Register 2037
14-164. CSI2_CTRL Register 2039
14-165. CSI2_GNQ Register 2042
14-166. CSI2_COMPLEXIO_CFG1 Register 2044
14-167. CSI2_COMPLEXIO_IRQSTATUS Register 2047
14-168. CSI2_COMPLEXIO_IRQENABLE Register 2051
14-169. CSI2_CLK_CTRL Register 2054
14-170. CSI2_TIMING1 Register 2056
14-171. CSI2_TIMING2 Register 2058
14-172. CSI2_VM_TIMING1 Register 2060
14-173. CSI2_VM_TIMING2 Register 2061
14-174. CSI2_VM_TIMING3 Register 2062
14-175. CSI2_CLK_TIMING Register 2063
14-176. CSI2_TX_FIFO_VC_SIZE Register 2064
14-177. CSI2_RX_FIFO_VC_SIZE Register 2065
14-178. CSI2_COMPLEXIO_CFG2 Register 2066
14-179. CSI2_RX_FIFO_VC_FULLNESS Register 2070
14-180. CSI2_VM_TIMING4 Register 2071
14-181. CSI2_TX_FIFO_VC_EMPTINESS Register 2072
14-182. CSI2_VM_TIMING5 Register 2073
14-183. CSI2_VM_TIMING6 Register 2074
14-184. CSI2_VM_TIMING7 Register 2075
14-185. CSI2_STOPCLK_TIMING Register 2076
14-186. CSI2_CTRL2 Register 2077
14-187. CSI2_VM_TIMING8 Register 2079
14-188. CSI2_TE_HSYNC_WIDTH_0 to CSI2_TE_HSYNC_WIDTH_1 Register 2080
14-189. CSI2_TE_VSYNC_WIDTH_0 to CSI2_TE_VSYNC_WIDTH_1 Register 2081
14-190. CSI2_TE_HSYNC_NUMBER_0 to CSI2_TE_HSYNC_NUMBER_1 Register 2082
14-191. CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register 2083
14-192. CSI2_VC_TE_0 to CSI2_VC_TE_3 Register 2087
14-193. CSI2_VC_LONG_PACKET_HEADER_0 to CSI2_VC_LONG_PACKET_HEADER_3 Register 2088
14-194. CSI2_VC_LONG_PACKET_PAYLOAD_0 to CSI2_VC_LONG_PACKET_PAYLOAD_3 Register 2089
14-195. CSI2_VC_SHORT_PACKET_HEADER_0 to CSI2_VC_SHORT_PACKET_HEADER_3 Register 2090
14-196. CSI2_VC_IRQSTATUS_0 to CSI2_VC_IRQSTATUS_3 Register 2091
14-197. CSI2_VC_IRQENABLE_0 to CSI2_VC_IRQENABLE_3 Register 2093
14-198. REGISTER0 Register 2096
14-199. REGISTER1 Register 2097
14-200. REGISTER2 Register 2098
14-201. REGISTER3 Register 2099
14-202. REGISTER4 Register 2100
14-203. REGISTER5 Register 2101
14-204. REGISTER6 Register 2102
14-205. REGISTER7 Register 2103
14-206. REGISTER8 Register 2104
14-207. REGISTER9 Register 2105
14-208. REGISTER10 Register 2107
14-209. REGISTER11 Register 2108

14-210. REGISTER12 Register 2109
14-211. REGISTER13 Register 2110
14-212. REGISTER14 Register 2111
14-213. REGISTER15 Register 2112
16-1. RTI Block Diagram 2117
16-2. Counter Block Diagram 2118
16-3. Compare Unit Block Diagram (shows only 1 of 4 blocks for simplification) 2120
16-4. Digital Watchdog 2120
16-5. DWD Operation 2121
16-6. Digital Windowed Watchdog Timing Example 2122
16-7. Digital Windowed Watchdog Operation Example (25% Window) 2122
16-8. RTI Global Control Register (RTIGCTRL) [offset = 00] 2125
16-9. RTI Timebase Control Register (RTITBCTRL) [offset = 04h] 2126
16-10. RTI Capture Control Register (RTICAPCTRL) [offset = 08h] 2127
16-11. RTI Compare Control Register (RTICOMPCTRL) [offset = 0Ch] 2128
16-12. RTI Free Running Counter 0 Register (RTIFRC0) [offset = 10h] 2129
16-13. RTI Up Counter 0 Register (RTIUC0) [offset = 14h] 2129
16-14. RTI Compare Up Counter 0 Register (RTICPUC0) [offset = 18h] 2130
16-15. RTI Capture Free Running Counter 0 Register (RTICAFRC0) [offset = 20h] 2130
16-16. RTI Capture Up Counter 0 Register (RTICAUC0) [offset = 24h] 2131
16-17. RTI Free Running Counter 1 Register (RTIFRC1) [offset = 30h] 2131
16-18. RTI Up Counter 1 Register (RTIUC1) [offset = 34h] 2132
16-19. RTI Compare Up Counter 1 Register (RTICPUC1) [offset = 38h] 2133
16-20. RTI Capture Free Running Counter 1 Register (RTICAFRC1) [offset = 40h] 2134
16-21. RTI Capture Up Counter 1 Register (RTICAUC1) [offset = 44h] 2134
16-22. RTI Compare 0 Register (RTICOMP0) [offset = 50h] 2135
16-23. RTI Update Compare 0 Register (RTIUDCP0) [offset = 54h] 2135
16-24. RTI Compare 1 Register (RTICOMP1) [offset = 58h] 2136
16-25. RTI Update Compare 1 Register (RTIUDCP1) [offset = 5Ch] 2136
16-26. RTI Compare 2 Register (RTICOMP2) [offset = 60h] 2137
16-27. RTI Update Compare 2 Register (RTIUDCP2) [offset = 64h] 2137
16-28. RTI Compare 3 Register (RTICOMP3) [offset = 68h] 2138
16-29. RTI Update Compare 3 Register (RTIUDCP3) [offset = 6Ch] 2138
16-30. RTI Timebase Low Compare Register (RTITBLCOMP) [offset = 70h] 2139
16-31. RTI Timebase High Compare Register (RTITBHCOMP) [offset = 74h] 2139
16-32. RTI Set Interrupt Control Register (RTISETINTENA) [offset = 80h] 2140
16-33. RTI Clear Interrupt Control Register (RTICLEARINTENA) [offset = 84h] 2142
16-34. RTI Interrupt Flag Register (RTIINTFLAG) [offset = 88h] 2144
16-35. Digital Watchdog Control Register (RTIDWDCTRL) [offset = 90h] 2145
16-36. Digital Watchdog Preload Register (RTIDWDPRLD) [offset = 94h] 2146
16-37. Watchdog Status Register (RTIWDSTATUS) [offset = 98h] 2147
16-38. RTI Watchdog Key Register (RTIDWDKEY) [offset = 9Ch] 2148
16-39. RTI Watchdog Down Counter Register (RTIDWDCNTR) [offset = A0h] 2149
16-40. Digital Windowed Watchdog Reaction Control (RTIWWDRXNCTRL) [offset = A4h] 2149
16-41. Digital Windowed Watchdog Window Size Control (RTIWWDSIZECTRL) [offset = A8h] 2150
16-42. RTI Compare Interrupt Clear Enable Register (RTIINTCLRENABLE) [offset = ACh] 2151
16-43. RTI Compare 0 Clear Register (RTICMP0CLR) [offset = B0h] 2152
16-44. RTI Compare 1 Clear Register (RTICMP1CLR) [offset = B4h] 2152
16-45. RTI Compare 2 Clear Register (RTICMP2CLR) [offset = B8h] 2153

16-46. RTI Compare 3 Clear Register (RTICMP3CLR) [offset = BCh] 2153
17-1. I/O Function Quick Start Flow Chart 2155
17-2. Interrupt Generation Function Quick Start Flow Chart 2156
17-3. GIO Module Diagram 2157
17-4. GIO Block Diagram 2159
17-5. GIOGCR Register 2163
17-6. GIOINTDET Register 2165
17-7. GIOPOL Register 2166
17-8. GIOENASET Register 2167
17-9. GIOENACLR Register 2168
17-10. GIOLVLSET Register 2169
17-11. GIOLVLCLR Register 2170
17-12. GIOFLG Register 2171
17-13. GIOOFFA Register 2172
17-14. GIOOFFB Register 2173
17-15. GIOEMUA Register 2174
17-16. GIOEMUB Register 2175
17-17. GIODIRA Register 2176
17-18. GIODINA Register 2177
17-19. GIODOUTA Register 2178
17-20. GIOSETA Register 2179
17-21. GIOCLRA Register 2180
17-22. GIOPDRA Register 2181
17-23. GIOPULDISA Register 2182
17-24. GIOPSLA Register 2183
17-25. GIODIRB Register 2184
17-26. GIODINB Register 2185
17-27. GIODOUTB Register 2186
17-28. GIOSETB Register 2187
17-29. GIOCLRB Register 2188
17-30. GIOPDRB Register 2189
17-31. GIOPULDISB Register 2190
17-32. GIOPSLB Register 2191
17-33. GIODIRC Register 2192
17-34. GIODINC Register 2193
17-35. GIODOUTC Register 2194
17-36. GIOSETC Register 2195
17-37. GIOCLRC Register 2196
17-38. GIOPDRC Register 2197
17-39. GIOPULDISC Register 2198
17-40. GIOPSLC Register 2199
17-41. GIODIRD Register 2200
17-42. GIODIND Register 2201
17-43. GIODOUTD Register 2202
17-44. GIOSETD Register 2203
17-45. GIOCLRD Register 2204
17-46. GIOPDRD Register 2205
17-47. GIOPULDISD Register 2206
17-48. GIOPSLD Register 2207

17-49. GIODIRE Register 2208
17-50. GIODINE Register 2209
17-51. GIODOUTE Register 2210
17-52. GIOSETE Register 2211
17-53. GIOCLRE Register 2212
17-54. GIOPDRE Register 2213
17-55. GIOPULDISE Register 2214
17-56. GIOPSLE Register 2215
17-57. GIODIRF Register 2216
17-58. GIODINF Register 2217
17-59. GIODOUTF Register 2218
17-60. GIOSETF Register 2219
17-61. GIOCLRF Register 2220
17-62. GIOPDRF Register 2221
17-63. GIOPULDISF Register 2222
17-64. GIOPSLF Register 2223
17-65. GIODIRG Register 2224
17-66. GIODING Register 2225
17-67. GIODOUTG Register 2226
17-68. GIOSETG Register 2227
17-69. GIOCLRG Register 2228
17-70. GIOPDRG Register 2229
17-71. GIOPULDISG Register 2230
17-72. GIOPSLG Register 2231
17-73. GIODIRH Register 2232
17-74. GIODINH Register 2233
17-75. GIODOUTH Register 2234
17-76. GIOSETH Register 2235
17-77. GIOCLRH Register 2236
17-78. GIOPDRH Register 2237
17-79. GIOPULDISH Register 2238
17-80. GIOPSLH Register 2239
17-81. GIOSRCA Register 2240
17-82. GIOSRCB Register 2241
17-83. GIOSRCC Register 2242
17-84. GIOSRCD Register 2243
17-85. GIOSRCE Register 2244
17-86. GIOSRCF Register 2245
17-87. GIOSRCG Register 2246
17-88. GIOSRCH Register 2247
18-1. Mailbox Block Diagram 2249
18-2. INT_MASK Register 2252
18-3. INT_MASK_SET Register 2253
18-4. INT_MASK_CLR Register 2254
18-5. INT_STS_CLR Register 2255
18-6. INT_ACK Register 2256
18-7. INT_TRIG Register 2257
18-8. INT_STS_MASKED Register 2258
18-9. INT_STS_RAW Register 2259

19-1. DMM Block Diagram 2261
19-2. Trace Mode Packet Format 2263
19-3. Direct Data Mode Packet Format 2263
19-4. Packet Sync Signal Example 2265
19-5. Example Single Packet Transmission 2265
19-6. Interrupt Structure 2266
19-7. DMM Global Control Register (DMMGLBCTRL) [offset = 00h] 2268
19-8. DMM Interrupt Set Register (DMMINTSET) [offset = 04h] 2270
19-9. DMM Interrupt Clear Register (DMMINTCLR) [offset = 08h] 2274
19-10. DMM Interrupt Level Register (DMMINTLVL) [offset = 0Ch] 2279
19-11. DMM Interrupt Flag Register (DMMINTFLG) [offset = 10h] 2281
19-12. DMM Interrupt Offset 1 Register (DMMOFF1) [offset = 14h] 2285
19-13. DMM Interrupt Offset 2 Register (DMMOFF2) [offset = 18h] 2286
19-14. DMM Direct Data Mode Destination Register (DMMDDMDEST) [offset = 1Ch] 2287
19-15. DMM Direct Data Mode Blocksize Register (DMMDDMBL) [offset = 20h] 2287
19-16. DMM Direct Data Mode Pointer Register (DMMDDMPT) [offset = 24h] 2288
19-17. DMM Direct Data Mode Interrupt Pointer Register (DMMINTPT) [offset = 28h] 2288
19-18. DMM Destination x Region 1 (DMMDESTxREG1) [offset = 2Ch, 3Ch, 4Ch, 5Ch] 2289
19-19. DMM Destination x Blocksize 1 (DMMDESTxBL1) [offset = 30h, 40h, 50h, 60h] 2290
19-20. DMM Destination x Region 2 (DMMDESTxREG2) [offset = 34h, 44h, 54h, 64h] 2291
19-21. DMM Destination x Blocksize 2 (DMMDESTxBL2) [offset = 38h, 48h, 58h, 68h] 2292
19-22. DMM Pin Control 0 (DMMPC0) [offset = 6Ch] 2293
19-23. DMM Pin Control 1 (DMMPC1) [offset = 70h] 2294
19-24. DMM Pin Control 2 (DMMPC2) [offset = 74h] 2296
19-25. DMM Pin Control 3 (DMMPC3) [offset = 78h] 2297
19-26. DMM Pin Control 4 (DMMPC4) [offset = 7Ch] 2298
19-27. DMM Pin Control 5 (DMMPC5) [offset = 80h] 2300
19-28. DMM Pin Control 6 (DMMPC6) [offset = 84h] 2301
19-29. DMM Pin Control 7 (DMMPC7) [offset = 88h] 2303
19-30. DMM Pin Control 8 (DMMPC8) [offset = 8Ch] 2304
20-1. ePWM Module 2307
20-2. Time-Base Submodule Block Diagram 2313
20-3. Time-Base Submodule Signals and Registers 2314
20-4. Time-Base Frequency and Period 2316
20-5. Time-Base Counter Synchronization Scheme 2317
20-6. Time-Base Up-Count Mode Waveforms 2319
20-7. Time-Base Down-Count Mode Waveforms 2320
20-8. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event 2320
20-9. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event 2321
20-10. Counter-Compare Submodule 2321
20-11. Detailed View of the Counter-Compare Submodule 2322
20-12. Counter-Compare Event Waveforms in Up-Count Mode 2324
20-13. Counter-Compare Events in Down-Count Mode 2324
20-14. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down On Synchronization Event 2325
20-15. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up On Synchronization Event 2325
20-16. Action-Qualifier Submodule 2326
20-17. Action-Qualifier Submodule Inputs and Outputs 2327

20-18. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs 2328
20-19. Up-Down-Count Mode Symmetrical Waveform 2331
20-20. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High 2332
20-21. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low 2333
20-22. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA 2334
20-23. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low 2336
20-24. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary 2337
20-25. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low 2338
20-26. Dead_Band Submodule 2339
20-27. Configuration Options for the Dead-Band Submodule 2340
20-28. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%) 2342
20-29. PWM-Chopper Submodule 2344
20-30. PWM-Chopper Submodule Operational Details 2345
20-31. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only 2345
20-32. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses 2346
20-33. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses 2347
20-34. Trip-Zone Submodule 2348
20-35. Trip-Zone Submodule Mode Control Logic 2352
20-36. Trip-Zone Submodule Interrupt Logic 2353
20-37. Event-Trigger Submodule 2354
20-38. Event-Trigger Submodule Inter-Connectivity of ADC Start of Conversion 2355
20-39. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs 2356
20-40. Event-Trigger Interrupt Generator 2358
20-41. Event-Trigger SOCA Pulse Generator 2358
20-42. Event-Trigger SOCB Pulse Generator 2359
20-43. Digital-Compare Submodule High-Level Block Diagram 2359
20-44. DCAEVT1 Event Triggering 2362
20-45. DCAEVT2 Event Triggering 2362
20-46. DCBEVT1 Event Triggering 2363
20-47. DCBEVT2 Event Triggering 2363
20-48. Event Filtering 2364
20-49. Blanking Window Timing Diagram 2365
20-50. Simplified ePWM Module 2366
20-51. EPWM1 Configured as a Typical Master, EPWM2 Configured as a Slave 2367
20-52. Control of Four Buck Stages. Here $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$ 2368
20-53. Buck Waveforms for (Note: Only three bucks shown here) 2369
20-54. Control of Four Buck Stages. (Note: $F_{PWM2} = N \times F_{PWM1}$) 2371
20-55. Buck Waveforms for (Note: $F_{PWM2} = F_{PWM1)}$) 2372
20-56. Control of Two Half-H Bridge Stages ($F_{PWM2} = N \times F_{PWM1}$) 2374
20-57. Half-H Bridge Waveforms for (Note: Here $F_{PWM2} = F_{PWM1}$) 2375
20-58. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control 2377
20-59. 3-Phase Inverter Waveforms for (Only One Inverter Shown) 2378
20-60. Configuring Two PWM Modules for Phase Control 2379
20-61. Timing Waveforms Associated With Phase Control Between 2 Modules 2380

20-62. TBCTL_TBSTS Register 2383
20-63. TBPHS Register 2385
20-64. TBCTR_TBPRD Register 2386
20-65. CMPCTL Register 2387
20-66. CMPA Register 2389
20-67. CMPB_AQCTLA Register 2390
20-68. AQCTLB_AQSFRC Register 2392
20-69. AQCSFRC_DBCTL Register 2394
20-70. DBRED_DBFED Register 2396
20-71. TZSEL_TZDCSEL Register 2397
20-72. TZCTL_TZEINT Register 2399
20-73. TZFLG_TZCLR Register 2401
20-74. TZFRC_ETSEL Register 2403
20-75. ETPS_ETFLG Register 2405
20-76. ETCLR_ETFRC Register 2407
20-77. PCCTL Register 2408
20-78. Reserved1 Register 2409
20-79. Reserved2 Register 2410
20-80. Reserved3 Register 2411
20-81. Reserved4 Register 2412
20-82. Reserved5 Register 2413
20-83. Reserved6 Register 2414
20-84. Reserved7 Register 2415
20-85. Reserved8 Register 2416
20-86. DCTRIPSEL_DCACTL Register 2417
20-87. DCBCTL_DCFCTL Register 2419
20-88. DCCAPCTL_DCFOFFSET Register 2420
20-89. DCFOFFSETCNT_DCFWINDOW Register 2421
20-90. DCFWINDOWCNT_DCCAP Register 2422
21-1. DCAN Block Diagram 2425
21-2. Bit Timing 2427
21-3. CAN Bit-timing Configuration 2432
21-4. Structure of a Message Object 2434
21-5. Message RAM Representation in Debug/Suspend Mode 2437
21-6. Message RAM Representation in RAM Direct Access Mode 2437
21-7. ECC RAM Representation 2438
21-8. Data Transfer Between IF1 / IF2 Registers and Message RAM 2440
21-9. Initialization of a Transmit Object 2442
21-10. Initialization of a Single Receive Object for Data Frames 2442
21-11. Initialization of a Single Receive Object for Remote Frames 2443
21-12. CPU Handling of a FIFO Buffer (Interrupt Driven) 2448
21-13. CAN Interrupt Topology 1 2451
21-14. CAN Interrupt Topology 2 2452
21-15. Local Power Down Mode Flow Diagram 2454
21-16. CAN Core in Silent Mode 2455
21-17. CAN Core in Loop Back Mode 2456
21-18. CAN Core in External Loop Back Mode 2457
21-19. CAN Core in Loop Back Combined with Silent Mode 2458
21-20. CTL Register 2464

21-21. ES Register 2467
21-22. ERRC Register 2469
21-23. BTR Register 2470
21-24. INT Register 2471
21-25. TEST Register 2472
21-26. Reserved_1 Register 2473
21-27. PERR Register 2474
21-28. DCAN_REV_ID Register 2475
21-29. ECCDIAG Register 2476
21-30. ECCDIAG_STAT Register 2477
21-31. ECC_CS Register 2478
21-32. ECC_SERR Register 2479
21-33. TTCAN1 Register 2480
21-34. TTCAN2 Register 2481
21-35. TTCAN3 Register 2482
21-36. TTCAN4 Register 2483
21-37. TTCAN5 Register 2484
21-38. TTCAN6 Register 2485
21-39. TTCAN7 Register 2486
21-40. TTCAN8 Register 2487
21-41. TTCAN9 Register 2488
21-42. TTCAN10 Register 2489
21-43. TTCAN11 Register 2490
21-44. TTCAN12 Register 2491
21-45. TTCAN13 Register 2492
21-46. TTCAN14 Register 2493
21-47. TTCAN15 Register 2494
21-48. TTCAN16 Register 2495
21-49. TTCAN17 Register 2496
21-50. TTCAN18 Register 2497
21-51. TTCAN19 Register 2498
21-52. ABOTR Register 2499
21-53. TXRQ_X Register 2500
21-54. TXRQ12 Register 2501
21-55. TXRQ34 Register 2502
21-56. TXRQ56 Register 2503
21-57. TXRQ78 Register 2504
21-58. NWDAT_X Register 2505
21-59. NWDAT12 Register 2506
21-60. NWDAT34 Register 2507
21-61. NWDAT56 Register 2508
21-62. NWDAT78 Register 2509
21-63. INTPND_X Register 2510
21-64. INTPND12 Register 2511
21-65. INTPND34 Register 2512
21-66. INTPND56 Register 2513
21-67. INTPND78 Register 2514
21-68. MSGVAL_X Register 2515
21-69. MSGVAL12 Register 2516

21-70. MSGVAL34 Register 2517
21-71. MSGVAL56 Register 2518
21-72. MSGVAL78 Register 2519
21-73. Reserved_2 Register 2520
21-74. INTMUX12 Register 2521
21-75. INTMUX34 Register 2522
21-76. INTMUX56 Register 2523
21-77. INTMUX78 Register 2524
21-78. Reserved_3 Register 2525
21-79. Reserved_4 Register 2526
21-80. Reserved_5 Register 2527
21-81. Reserved_6 Register 2528
21-82. Reserved_7 Register 2529
21-83. Reserved_8 Register 2530
21-84. IF1CMD Register 2531
21-85. IF1MSK Register 2534
21-86. IF1ARB Register 2535
21-87. IF1MCTL Register 2536
21-88. IF1DATA Register 2538
21-89. IF1DATB Register 2539
21-90. Reserved_9 Register 2540
21-91. Reserved_10 Register 2541
21-92. IF2CMD Register 2542
21-93. IF2MSK Register 2545
21-94. IF2ARB Register 2546
21-95. IF2MCTL Register 2547
21-96. IF2DATA Register 2549
21-97. IF2DATB Register 2550
21-98. Reserved_11 Register 2551
21-99. Reserved_12 Register 2552
21-100. IF3OBS Register 2553
21-101. IF3MSK Register 2555
21-102. IF3ARB Register 2556
21-103. IF3MCTL Register 2557
21-104. IF3DATA Register 2559
21-105. IF3DATB Register 2560
21-106. Reserved_13 Register 2561
21-107. Reserved_14 Register 2562
21-108. IF3UPD12 Register 2563
21-109. IF3UPD34 Register 2564
21-110. IF3UPD56 Register 2565
21-111. IF3UPD78 Register 2566
21-112. Reserved_15 Register 2567
21-113. Reserved_16 Register 2568
21-114. Reserved_17 Register 2569
21-115. Reserved_18 Register 2570
21-116. Reserved_19 Register 2571
21-117. Reserved_20 Register 2572
21-118. Reserved_21 Register 2573

21-119. Reserved_22 Register 2574
21-120. Reserved_23 Register 2575
21-121. Reserved_24 Register 2576
21-122. Reserved_25 Register 2577
21-123. Reserved_26 Register 2578
21-124. Reserved_27 Register 2579
21-125. Reserved_28 Register 2580
21-126. Reserved_29 Register 2581
21-127. Reserved_30 Register 2582
21-128. Reserved_31 Register 2583
21-129. Reserved_32 Register 2584
21-130. Reserved_33 Register 2585
21-131. Reserved_34 Register 2586
21-132. Reserved_35 Register 2587
21-133. Reserved_36 Register 2588
21-134. Reserved_37 Register 2589
21-135. Reserved_38 Register 2590
21-136. Reserved_39 Register 2591
21-137. Reserved_40 Register 2592
21-138. Reserved_41 Register 2593
21-139. Reserved_42 Register 2594
21-140. TIOC Register 2595
21-141. RIOC Register 2597
22-1. MCAN Block Diagram 2601
22-2. CAN Bit Timing 2605
22-3. Transmitter Delay Measurement 2606
22-4. Connection of Signals in Bus Monitoring Mode 2607
22-5. Internal Loop Back Mode 2610
22-6. External Timestamp Counter Interrupt 2611
22-7. Standard Message ID Filter Path 2616
22-8. Extended Message ID Filter Path 2617
22-9. Rx FIFO Status 2618
22-10. Rx FIFO Overflow Handling 2619
22-11. Mixed Dedicated Tx Buffers / Tx FIFO (example) 2622
22-12. Mixed Dedicated Tx Buffers / Tx Queue (example) 2623
22-13. Message RAM Configuration 2625
22-14. Rx Buffer / Rx FIFO Element Structure 2625
22-15. Tx Buffer Element Structure 2627
22-16. Tx Event FIFO Element Structure 2629
22-17. Standard Message ID Filter Element Structure 2630
22-18. Extended Message ID Filter Element Structure 2632
22-19. MCANSS_PID Register 2637
22-20. MCANSS_CTRL Register 2638
22-21. MCANSS_STAT Register 2639
22-22. MCANSS_ICS Register 2640
22-23. MCANSS_IRS Register 2641
22-24. MCANSS_IECS Register 2642
22-25. MCANSS_IE Register 2643
22-26. MCANSS_IES Register 2644

22-27. MCANSS_EOI Register 2645
22-28. MCANSS_EXT_TS_PS Register 2646
22-29. MCANSS_EXT_TS_USIC Register 2647
22-30. MCANSS_ECC_EOI Register 2648
22-31. MCAN_CREL Register 2649
22-32. MCAN_ENDN Register 2650
22-33. MCAN_CUST Register 2651
22-34. MCAN_DBTP Register 2652
22-35. MCAN_TEST Register 2653
22-36. MCAN_RWD Register 2654
22-37. MCAN_CCCR Register 2655
22-38. MCAN_NBTP Register 2656
22-39. MCAN_TSCC Register 2657
22-40. MCAN_TSCV Register 2658
22-41. MCAN_TOCC Register 2659
22-42. MCAN_TOCV Register 2660
22-43. MCAN_RES00 Register 2661
22-44. MCAN_RES01 Register 2662
22-45. MCAN_RES02 Register 2663
22-46. MCAN_RES03 Register 2664
22-47. MCAN_ECR Register 2665
22-48. MCAN_PSR Register 2666
22-49. MCAN_TDCR Register 2667
22-50. MCAN_RES04 Register 2668
22-51. MCAN_IR Register 2669
22-52. MCAN_IE Register 2671
22-53. MCAN_ILS Register 2673
22-54. MCAN_ILE Register 2675
22-55. MCAN_RES05 Register 2676
22-56. MCAN_RES06 Register 2677
22-57. MCAN_RES07 Register 2678
22-58. MCAN_RES08 Register 2679
22-59. MCAN_RES09 Register 2680
22-60. MCAN_RES10 Register 2681
22-61. MCAN_RES11 Register 2682
22-62. MCAN_RES12 Register 2683
22-63. MCAN_GFC Register 2684
22-64. MCAN_SIDFC Register 2685
22-65. MCAN_XIDFC Register 2686
22-66. MCAN_RES13 Register 2687
22-67. MCAN_XIDAM Register 2688
22-68. MCAN_HPMS Register 2689
22-69. MCAN_NDAT1 Register 2690
22-70. MCAN_NDAT2 Register 2691
22-71. MCAN_RXF0C Register 2692
22-72. MCAN_RXF0S Register 2693
22-73. MCAN_RXF0A Register 2694
22-74. MCAN_RXBC Register 2695
22-75. MCAN_RXF1C Register 2696

22-76. MCAN_RXF1S Register 2697
22-77. MCAN_RXF1A Register 2698
22-78. MCAN_RXESC Register 2699
22-79. MCAN_TXBC Register 2700
22-80. MCAN_TXFQS Register 2701
22-81. MCAN_TXESC Register 2702
22-82. MCAN_TXBRP Register 2703
22-83. MCAN_TXBAR Register 2704
22-84. MCAN_TXBCR Register 2705
22-85. MCAN_TXBTO Register 2706
22-86. MCAN_TXBCF Register 2707
22-87. MCAN_TXBTIE Register 2708
22-88. MCAN_TXBCIE Register 2709
22-89. MCAN_RES14 Register 2710
22-90. MCAN_RES15 Register 2711
22-91. MCAN_TXEFC Register 2712
22-92. MCAN_TXEFS Register 2713
22-93. MCAN_TXEFA Register 2714
22-94. MCAN_RES16 Register 2715
22-95. MCANSS_ECC_AGGR_REVISION Register 2717
22-96. MCANSS_ECC_VECTOR Register 2718
22-97. MCANSS_ECC_MISC_STATUS Register 2719
22-98. MCANSS_ECC_WRAP_REVISION Register 2720
22-99. MCANSS_ECC_CONTROL Register 2721
22-100. MCANSS_ECC_ERR_CTRL1 Register 2722
22-101. MCANSS_ECC_ERR_CTRL2 Register 2723
22-102. MCANSS_ECC_ERR_STAT1 Register 2724
22-103. MCANSS_ECC_ERR_STAT2 Register 2725
22-104. MCANSS_ECC_SEC_EOI_REG Register 2726
22-105. MCANSS_ECC_SEC_STATUS_REG0 Register 2727
22-106. MCANSS_ECC_SEC_ENABLE_SET_REG0 Register 2728
22-107. MCANSS_ECC_SEC_ENABLE_CLR_REG0 Register 2729
22-108. MCANSS_ECC_DED_EOI_REG Register 2730
22-109. MCANSS_ECC_DED_STATUS_REG0 Register 2731
22-110. MCANSS_ECC_DED_ENABLE_SET_REG0 Register 2732
22-111. MCANSS_ECC_DED_ENABLE_CLR_REG0 Register 2733
22-112. MCAN Block Diagram 2735
22-113. CAN Bit Timing 2739
22-114. Transmitter Delay Measurement 2740
22-115. Connection of Signals in Bus Monitoring Mode 2741
22-116. Internal Loop Back Mode 2744
22-117. External Timestamp Counter Interrupt 2745
22-118. Standard Message ID Filter Path 2750
22-119. Extended Message ID Filter Path 2751
22-120. Rx FIFO Status 2752
22-121. Rx FIFO Overflow Handling 2753
22-122. Mixed Dedicated Tx Buffers / Tx FIFO (example) 2756
22-123. Mixed Dedicated Tx Buffers / Tx Queue (example) 2757
22-124. Message RAM Configuration 2759

22-125. Rx Buffer / Rx FIFO Element Structure 2759
22-126. Tx Buffer Element Structure 2761
22-127. Tx Event FIFO Element Structure 2763
22-128. Standard Message ID Filter Element Structure 2764
22-129. Extended Message ID Filter Element Structure 2766
22-130. SS_PID Register 2771
22-131. SS_CTRL Register 2772
22-132. SS_STAT Register 2773
22-133. SS_ICS Register 2774
22-134. SS_IRS Register 2775
22-135. SS_IECS Register 2776
22-136. SS_IE Register 2777
22-137. SS_IES Register 2778
22-138. SS_EOI Register 2779
22-139. SS_EXT_TS_PS Register 2780
22-140. SS_EXT_TS_USIC Register 2781
22-141. CREL Register 2782
22-142. ENDN Register 2783
22-143. CUST Register 2784
22-144. DBTP Register 2785
22-145. TEST Register 2786
22-146. RWD Register 2787
22-147. CCCR Register 2788
22-148. NBTP Register 2789
22-149. TSCC Register 2790
22-150. TSCV Register 2791
22-151. TOCC Register 2792
22-152. TOCV Register 2793
22-153. RES00 Register 2794
22-154. RES01 Register 2795
22-155. RES02 Register 2796
22-156. RES03 Register 2797
22-157. ECR Register 2798
22-158. PSR Register 2799
22-159. TDCR Register 2800
22-160. RES04 Register 2801
22-161. IR Register 2802
22-162. IE Register 2804
22-163. ILS Register 2806
22-164. ILE Register 2808
22-165. RES05 Register 2809
22-166. RES06 Register 2810
22-167. RES07 Register 2811
22-168. RES08 Register 2812
22-169. RES09 Register 2813
22-170. RES10 Register 2814
22-171. RES11 Register 2815
22-172. RES12 Register 2816
22-173. GFC Register 2817

22-174. SIDFC Register 2818
22-175. XIDFC Register 2819
22-176. RES13 Register 2820
22-177. XIDAM Register 2821
22-178. HPMS Register 2822
22-179. NDAT1 Register 2823
22-180. NDAT2 Register 2824
22-181. RXF0C Register 2825
22-182. RXF0S Register 2826
22-183. RXF0A Register 2827
22-184. RXBC Register 2828
22-185. RXF1C Register 2829
22-186. RXF1S Register 2830
22-187. RXF1A Register 2831
22-188. RXESC Register 2832
22-189. TXBC Register 2833
22-190. TXFQS Register 2834
22-191. TXESC Register 2835
22-192. TXBRP Register 2836
22-193. TXBAR Register 2837
22-194. TXBCR Register 2838
22-195. TXBTO Register 2839
22-196. TXBCF Register 2840
22-197. TXBTIE Register 2841
22-198. TXBCIE Register 2842
22-199. RES14 Register 2843
22-200. RES15 Register 2844
22-201. TXEFC Register 2845
22-202. TXEFS Register 2846
22-203. TXEFA Register 2847
22-204. RES16 Register 2848
22-205. REV Register 2850
22-206. VECTOR Register 2851
22-207. STAT Register 2852
22-208. CTRL Register 2853
22-209. ERR_CTRL1 Register 2854
22-210. ERR_CTRL2 Register 2855
22-211. ERR_STAT1 Register 2856
22-212. ERR_STAT2 Register 2857
22-213. ERR_STAT3 Register 2858
22-214. SEC_EOI_REG Register 2859
22-215. SEC_STATUS_REG0 Register 2860
22-216. SEC_ENABLE_SET_REG0 Register 2861
22-217. SEC_ENABLE_CLR_REG0 Register 2862
22-218. DED_EOI_REG Register 2863
22-219. DED_STATUS_REG0 Register 2864
22-220. DED_ENABLE_SET_REG0 Register 2865
22-221. DED_ENABLE_CLR_REG0 Register 2866
22-222. AGGR_ENABLE_SET Register 2867

22-223. AGGR_ENABLE_CLR Register 2868
22-224. AGGR_STATUS_SET Register 2869
22-225. AGGR_STATUS_CLR Register 2870
23-1. SPI Functional Logic Diagram 2875
23-2. MibSPI Functional Logic Diagram 2876
23-3. DMA Channel and Request Line (Logical) Structure in Multi-buffer Mode 2878
23-4. TG Interrupt Structure 2880
23-5. SPIFLG Interrupt Structure 2880
23-6. SPI Three-Pin Operation 2881
23-7. Operation with $\overline{\text{SPISCS}}$ 2882
23-8. Operation with $\overline{\text{SPIENA}}$ 2883
23-9. SPI Five-Pin Option with $\overline{\text{SPIENA}}$ and $\overline{\text{SPISCS}}$ 2884
23-10. Format for Transmitting an 12-Bit Word 2885
23-11. Format for Receiving an 10-Bit Word 2885
23-12. Clock Mode with Polarity = 0 and Phase = 0 2886
23-13. Clock Mode with Polarity = 0 and Phase = 1 2886
23-14. Clock Mode with Polarity = 1 and Phase = 0 2887
23-15. Clock Mode with Polarity = 1 and Phase = 1 2887
23-16. Five Bits per Character (5-Pin Option) 2888
23-17. Example: $t_{C2TDELAY}$= 8 VCLK Cycles 2889
23-18. Example: $t_{T2CDELAY}$= 4 VCLK Cycles 2890
23-19. Transmit-Data-Finished-to-ENA-Inactive-Timeout 2890
23-20. Chip-Select-Active-to-ENA-Signal-Active-Timeout 2891
23-21. Typical Diagram when a Buffer in Master is in CSHOLD Mode (SPI-SPI) 2892
23-22. Block Diagram Shift Register, MSB First 2894
23-23. Block Diagram Shift Register, LSB First 2894
23-24. 2-data Line Mode (Phase 0, Polarity 0) 2897
23-25. Two-Pin Parallel Mode Timing Diagram (Phase 0, Polarity 0) 2897
23-26. 4-Data Line Mode (Phase 0, Polarity 0) 2898
23-27. 4 Pins Parallel Mode Timing Diagram (Phase 0, Polarity 0) 2898
23-28. Eight-data Line Mode (Phase 0, Polarity 0) 2899
23-29. 8 Pins Parallel Mode Timing Diagram (Phase 0, Polarity 0) 2900
23-30. Multi-buffer in Slave Mode 2901
23-31. I/O Paths During I/O Loopback Modes 2906
23-32. SPIGCR0 Register 2913
23-33. SPIGCR1 Register 2914
23-34. SPIINT0 Register 2916
23-35. SPILVL Register 2918
23-36. SPIFLG Register 2919
23-37. SPIPC0 Register 2922
23-38. SPIPC1 Register 2924
23-39. SPIPC2 Register 2926
23-40. SPIPC3 Register 2927
23-41. SPIPC4 Register 2929
23-42. SPIPC5 Register 2931
23-43. SPIPC6 Register 2933
23-44. SPIDAT0 Register 2935
23-45. SPIDAT1 Register 2936
23-46. SPIBUF Register 2938

23-47. SPIEMU Register 2941
23-48. SPIDELAY Register 2943
23-49. SPIDEF Register 2945
23-50. SPIFMT0 Register 2946
23-51. SPIFMT1 Register 2948
23-52. SPIFMT2 Register 2950
23-53. SPIFMT3 Register 2952
23-54. TGINTVECT0 Register 2954
23-55. TGINTVECT1 Register 2956
23-56. SPIPC9 Register 2958
23-57. SPIPMCTRL Register 2959
23-58. MIBSPIE Register 2962
23-59. TGITENST Register 2964
23-60. TGITENCR Register 2965
23-61. TGITLVST Register 2966
23-62. TGITLVCR Register 2967
23-63. TGINTFLAG Register 2968
23-64. TICKCNT Register 2969
23-65. LTGPEND Register 2970
23-66. TG0CTRL Register 2971
23-67. TG1CTRL Register 2974
23-68. TG2CTRL Register 2977
23-69. TG3CTRL Register 2980
23-70. TG4CTRL Register 2983
23-71. TG5CTRL Register 2986
23-72. TG6CTRL Register 2989
23-73. TG7CTRL Register 2992
23-74. DMA0CTRL Register 2995
23-75. DMA1CTRL Register 2997
23-76. DMA2CTRL Register 2999
23-77. DMA3CTRL Register 3001
23-78. DMA4CTRL Register 3003
23-79. ICOUNT0 Register 3005
23-80. ICOUNT1 Register 3006
23-81. ICOUNT2 Register 3007
23-82. ICOUNT3 Register 3008
23-83. ICOUNT4 Register 3009
23-84. DMACNTLEN Register 3010
23-85. PAR_ECC_CTRL Register 3011
23-86. PAR_ECC_STAT Register 3012
23-87. UERRADDR1 Register 3013
23-88. UERRADDR0 Register 3014
23-89. RXOVRN_BUF_ADDR Register 3015
23-90. IOLPBKTSTCR Register 3016
23-91. EXTENDED_PRESCALE1 Register 3018
23-92. EXTENDED_PRESCALE2 Register 3020
23-93. ECCDIAG_CTRL Register 3021
23-94. ECCDIAG_STAT Register 3022
23-95. SBERRADDR1 Register 3023

23-96. SBERRADDR0 Register 3024
23-97. SPIREV Register 3025
23-98. SPIGCR0 Register 3029
23-99. SPIGCR1 Register 3030
23-100. SPIINT0 Register 3032
23-101. SPILVL Register 3034
23-102. SPIFLG Register 3035
23-103. SPIPC0 Register 3038
23-104. SPIPC1 Register 3040
23-105. SPIPC2 Register 3042
23-106. SPIPC3 Register 3043
23-107. SPIPC4 Register 3045
23-108. SPIPC5 Register 3047
23-109. SPIPC6 Register 3049
23-110. SPIDAT0 Register 3051
23-111. SPIDAT1 Register 3052
23-112. SPIBUF Register 3054
23-113. SPIEMU Register 3057
23-114. SPIDELAY Register 3059
23-115. SPIDEF Register 3061
23-116. SPIFMT0 Register 3062
23-117. SPIFMT1 Register 3064
23-118. SPIFMT2 Register 3066
23-119. SPIFMT3 Register 3068
23-120. TGINTVECT0 Register 3070
23-121. TGINTVECT1 Register 3072
23-122. SPIPC9 Register 3074
23-123. SPIPMCTRL Register 3075
23-124. MIBSPIE Register 3078
23-125. TGITENST Register 3080
23-126. TGITENCR Register 3081
23-127. TGITLVST Register 3082
23-128. TGITLVCR Register 3083
23-129. TGINTFLAG Register 3084
23-130. TICKCNT Register 3085
23-131. LTGPEND Register 3086
23-132. TG0CTRL Register 3087
23-133. TG1CTRL Register 3090
23-134. TG2CTRL Register 3093
23-135. TG3CTRL Register 3096
23-136. TG4CTRL Register 3099
23-137. TG5CTRL Register 3102
23-138. TG6CTRL Register 3105
23-139. TG7CTRL Register 3108
23-140. DMA0CTRL Register 3111
23-141. DMA1CTRL Register 3113
23-142. DMA2CTRL Register 3115
23-143. DMA3CTRL Register 3117
23-144. DMA4CTRL Register 3119

23-145. ICOUNT0 Register 3121
23-146. ICOUNT1 Register 3122
23-147. ICOUNT2 Register 3123
23-148. ICOUNT3 Register 3124
23-149. ICOUNT4 Register 3125
23-150. DMACNTLEN Register 3126
23-151. PAR_ECC_CTRL Register 3127
23-152. PAR_ECC_STAT Register 3128
23-153. UERRADDR1 Register 3129
23-154. UERRADDR0 Register 3130
23-155. RXOVRN_BUF_ADDR Register 3131
23-156. IOLPBKTSTCR Register 3132
23-157. EXTENDED_PRESCALE1 Register 3134
23-158. EXTENDED_PRESCALE2 Register 3136
23-159. ECCDIAG_CTRL Register 3137
23-160. ECCDIAG_STAT Register 3138
23-161. SBERRADDR1 Register 3139
23-162. SBERRADDR0 Register 3140
23-163. SPIREV Register 3141
23-164. Multi-Buffer RAM Configuration When Parity Check is Supported 3142
23-165. Multi-Buffer RAM Configuration When ECC Check is Supported 3142
23-166. Multi-Buffer RAM Transmit Data Register [offset = Base + 000-1FFh] 3144
23-167. Multi-buffer RAM Receive Buffer Register [offset = RAM Base + 200-3FFh] 3146
23-168. Memory-Map for Parity Locations During Normal and Test Mode While EXTENDED_BUF Mode is Disabled or the Feature is Not Implemented 3149
23-169. Memory-Map for Parity Locations During Normal and Test Mode While EXTENDED_BUF Mode is Enabled 3150
23-170. Example of Memory-Mapped Parity Locations During Test Mode 3151
23-171. Example of ECC Bit Locations During Test Mode 3152
23-172. SPI/MibSPI Pins During Master Mode 3-Pin Configuration 3153
23-173. SPI/MibSPI Pins During Master Mode 4-Pin with $\overline{\text{SPISCS}}$ Configuation 3153
23-174. SPI/MibSPI Pins During Master Mode in 4-Pin with $\overline{\text{SPIENA}}$ Configuration 3154
23-175. SPI/MibSPI Pins During Master/Slave Mode with 5-Pin Configuration 3154
23-176. SPI/MibSPI Pins During Slave Mode 3-Pin Configuration 3155
23-177. SPI/MibSPI Pins During Slave Mode in 4-Pin with $\overline{\text{SPIENA}}$ Configuration 3155
23-178. SPI/MibSPI Pins During Slave Mode in 5-Pin Configuration (Single Slave) 3155
23-179. SPI/MibSPI Pins During Slave Mode in 5-Pin Configuration (Single/Multi-Slave) 3155
24-1. QSPI Overview 3158
24-2. QSPI Block Diagram 3159
24-3. SPI_CLKGEN Block 3163
24-4. SPI Clock Modes 3164
24-5. Logical Representation of the QSPI Interrupt Generation Scheme 3165
24-6. PID Register 3169
24-7. SYSCONFIG Register 3170
24-8. INTR_STATUS_RAW_SET Register 3171
24-9. INTR_STATUS_ENABLED_CLEAR Register 3172
24-10. INTR_ENABLE_SET Register 3173
24-11. INTR_ENABLE_CLEAR Register 3174
24-12. INTC_EOI Register 3175

24-13. SPI_CLOCK_CNTRL Register 3176
24-14. SPI_DC Register 3177
24-15. SPI_CMD Register 3179
24-16. SPI_STATUS Register 3180
24-17. SPI_DATA Register 3181
24-18. SPI_SETUP0 Register 3182
24-19. SPI_SETUP1 Register 3183
24-20. SPI_SETUP2 Register 3184
24-21. SPI_SETUP3 Register 3185
24-22. SPI_SWITCH Register 3186
24-23. SPI_DATA1 Register 3187
24-24. SPI_DATA2 Register 3188
24-25. SPI_DATA3 Register 3189
25-1. Multiple I2C Modules Connection Diagram 3191
25-2. Simple I2C Block Diagram 3193
25-3. Clocking Diagram for the I2C Module 3194
25-4. Bit Transfer on the I2C Bus 3195
25-5. I2C Module START and STOP Conditions 3196
25-6. I2C Module Data Transfer 3196
25-7. I2C Module 7-Bit Addressing Format 3197
25-8. I2C Module 10-bit Addressing Format 3197
25-9. I2C Module 7-Bit Addressing Format with Repeated START 3197
25-10. I2C Module in Free Data Format 3198
25-11. Arbitration Procedure Between Two Master Transmitters 3201
25-12. Synchronization of Two I2C Clock Generators During Arbitration 3202
25-13. ICOAR Register 3207
25-14. ICIMR Register 3208
25-15. ICSTR Register 3209
25-16. ICCLKL Register 3211
25-17. ICCLKH Register 3212
25-18. ICCNT Register 3213
25-19. ICDRR Register 3214
25-20. ICSAR Register 3215
25-21. ICDXR Register 3216
25-22. ICMDR Register 3217
25-23. ICIVR Register 3220
25-24. ICEMDR Register 3221
25-25. ICPSC Register 3222
25-26. ICPID1 Register 3223
25-27. ICPID2 Register 3224
25-28. ICDMAC Register 3225
25-29. ICPFUNC Register 3226
25-30. ICPDIR Register 3227
25-31. ICPDIN Register 3228
25-32. ICPDOUT Register 3229
25-33. ICPDSET Register 3230
25-34. ICPDCLR Register 3231
25-35. ICPDRV Register 3232
25-36. Difference between Normal Operation and Backward Compatibility Mode 3233

26-1. Detailed SCI Block Diagram 3236
26-2. Typical SCI Data Frame Formats 3237
26-3. Asynchronous Communication Bit Timing 3238
26-4. Idle-Line Multiprocessor Communication Format 3240
26-5. Address-Bit Multiprocessor Communication Format 3241
26-6. General Interrupt Scheme 3242
26-7. Interrupt Generation for Given Flags 3243
26-8. SCI Global Control Register 0 (SCIGCR0) [offset = 00] 3251
26-9. SCI Global Control Register 1 (SCIGCR1) [offset = 04h] 3252
26-10. SCI Set Interrupt Register (SCISETINT) [offset = 0Ch] 3255
26-11. SCI Clear Interrupt Register (SCICLEARINT) [offset = 10h] 3257
26-12. SCI Set Interrupt Level Register (SCISETINTLVL) [offset = 14h] 3259
26-13. SCI Clear Interrupt Level Register (SCICLEARINTLVL) [offset = 18h] 3260
26-14. SCI Flags Register (SCIFLR) [offset = 1Ch] 3262
26-15. SCI Interrupt Vector Offset 0 (SCIINTVECT0) [offset = 20h] 3266
26-16. SCI Interrupt Vector Offset 1 (SCIINTVECT1) [offset = 24h] 3266
26-17. SCI Format Control Register (SCIFORMAT) [offset = 28h] 3267
26-18. Baud Rate Selection Register (BRS) [offset = 2Ch] 3268
26-19. Receiver Emulation Data Buffer (SCIED) [offset = 30h] 3269
26-20. Receiver Data Buffer (SCIRD) [offset = 34h] 3269
26-21. Transmit Data Buffer Register (SCITD) [offset = 38h] 3270
26-22. SCI Pin I/O Control Register 0 (SCIPIO0) [offset = 3Ch] 3270
26-23. SCI Pin I/O Control Register 1 (SCIPIO1) [offset = 40h] 3271
26-24. SCI Pin I/O Control Register 2 (SCIPIO2) [offset = 44h] 3272
26-25. SCI Pin I/O Control Register 3 (SCIPIO3) [offset = 48h] 3273
26-26. SCI Pin I/O Control Register 4 (SCIPIO4) [offset = 4Ch] 3274
26-27. SCI Pin I/O Control Register 5 (SCIPIO5) [offset = 50h] 3275
26-28. SCI Pin I/O Control Register 6 (SCIPIO6) [offset = 54h] 3276
26-29. SCI Pin I/O Control Register 7 (SCIPIO7) [offset = 58h] 3277
26-30. SCI Pin I/O Control Register 8 (SCIPIO8) [offset = 5Ch] 3277
26-31. Input/Output Error Enable Register (IODFTCTRL) [offset = 90h] 3278
26-32. GPIO Functionality 3280
27-1. Block Diagram of the 14xx Debug Subsystem Interconnect 3282
27-2. Block Diagram of the 16xx/18xx/68xx Debug Subsystem Interconnect 3283
27-3. 16xx/18xx/68xx Cross Trigger Subsystem 3284
28-1. DCC Operation 3286
28-2. Counter Relationship 3288
28-3. Clock1 Slower Than Clock0 - Results in an Error and Stops Counting 3288
28-4. Clock1 Faster Than Clock0 - Results in an Error and Stops Counting 3289
28-5. Clock1 Not Present - Results in an Error and Stops Counting 3289
28-6. Clock0 Not Present - Results in an Error and Stops Counting 3290
28-7. DCCGCTRL Register 3293
28-8. DCCREV Register 3294
28-9. DCCCNTSEED0 Register 3295
28-10. DCCVALIDSEED0 Register 3296
28-11. DCCCNTSEED1 Register 3297
28-12. DCCSTAT Register 3298
28-13. DCCCNT0 Register 3299
28-14. DCCVALID0 Register 3300

28-15. DCCCNT1 Register 3301
28-16. DCCCLKSSRC1 Register 3302
28-17. DCCCLKSSRC0 Register 3303
28-18. Block Diagram 3304
28-19. Interrupt Response Handling 3305
28-20. $\overline{\text{ERROR}}$ Pin Response Handling 3305
28-21. $\overline{\text{ERROR}}$ Pin Timing - Example 1 3307
28-22. $\overline{\text{ERROR}}$ Pin Timing - Example 2 3307
28-23. $\overline{\text{ERROR}}$ Pin Timing - Example 3 3307
28-24. $\overline{\text{ERROR}}$ Pin Timing - Example 4 3308
28-25. $\overline{\text{ERROR}}$ Pin Timing - Example 5 3308
28-26. ERROR Pin Timing - Example 6 3308
28-27. ESM Initialization 3309
28-28. ESMIEPSR1 Register 3311
28-29. ESMIEPCR1 Register 3312
28-30. ESMIESR1 Register 3313
28-31. ESMIECR1 Register 3314
28-32. ESMILSR1 Register 3315
28-33. ESMILCR1 Register 3316
28-34. ESMSR1 Register 3317
28-35. ESMSR2 Register 3318
28-36. ESMSR3 Register 3319
28-37. ESMEPSR Register 3320
28-38. ESMIOFFHR Register 3321
28-39. ESMIOFFLR Register 3322
28-40. ESMLTCR Register 3323
28-41. ESMLTCPR Register 3324
28-42. ESMEKR Register 3325
28-43. ESMSSR2 Register 3326
28-44. ESMIEPSR4 Register 3327
28-45. ESMIEPCR4 Register 3328
28-46. ESMIESR4 Register 3329
28-47. ESMIECR4 Register 3330
28-48. ESMILSR4 Register 3331
28-49. ESMILCR4 Register 3332
28-50. ESMSR4 Register 3333
28-51. ESMIEPSR7 Register 3334
28-52. ESMIEPCR7 Register 3335
28-53. ESMIESR7 Register 3336
28-54. ESMIECR7 Register 3337
28-55. ESMILSR7 Register 3338
28-56. ESMILCR7 Register 3339
28-57. ESMSR7 Register 3340
28-58. CRC Controller Block Diagram For One Channel 3341
28-59. Linear Feedback Shift Register (LFSR) 3343
28-60. AUTO Mode Using Hardware Timer Trigger 3346
28-61. AUTO Mode With Software CPU Trigger 3346
28-62. Semi-CPU Mode With Hardware Timer Trigger 3347
28-63. Timeout Example 1 3349

28-64. Timeout Example 2 3350
28-65. Timeout Example 3 3350
28-66. CRC_CTRL0 Register 3358
28-67. CRC_CTRL1 Register 3360
28-68. CRC_CTRL2 Register 3361
28-69. CRC_INTS Register 3362
28-70. CRC_INTR Register 3365
28-71. CRC_STATUS_REG Register 3368
28-72. CRC_INT_OFFSET_REG Register 3370
28-73. CRC_BUSY Register 3371
28-74. CRC_PCOUNT_REG1 Register 3372
28-75. CRC_SCOUNT_REG1 Register 3373
28-76. CRC_CURSEC_REG1 Register 3374
28-77. CRC_WDTOPLD1 Register 3375
28-78. CRC_BCTOPLD1 Register 3376
28-79. PSA_SIGREGL1 Register 3377
28-80. PSA_SIGREGH1 Register 3378
28-81. CRC_REGL1 Register 3379
28-82. CRC_REGH1 Register 3380
28-83. PSA_SECSIGREGL1 Register 3381
28-84. PSA_SECSIGREGH1 Register 3382
28-85. RAW_DATAREGL1 Register 3383
28-86. RAW_DATAREGH1 Register 3384
28-87. CRC_PCOUNT_REG2 Register 3385
28-88. CRC_SCOUNT_REG2 Register 3386
28-89. CRC_CURSEC_REG2 Register 3387
28-90. CRC_WDTOPLD2 Register 3388
28-91. CRC_BCTOPLD2 Register 3389
28-92. PSA_SIGREGL2 Register 3390
28-93. PSA_SIGREGH2 Register 3391
28-94. CRC_REGL2 Register 3392
28-95. CRC_REGH2 Register 3393
28-96. PSA_SECSIGREGL2 Register 3394
28-97. PSA_SECSIGREGH2 Register 3395
28-98. RAW_DATAREGL2 Register 3396
28-99. RAW_DATAREGH2 Register 3397
28-100. CRC_PCOUNT_REG3 Register 3398
28-101. CRC_SCOUNT_REG3 Register 3399
28-102. CRC_CURSEC_REG3 Register 3400
28-103. CRC_WDTOPLD3 Register 3401
28-104. CRC_BCTOPLD3 Register 3402
28-105. PSA_SIGREGL3 Register 3403
28-106. PSA_SIGREGH3 Register 3404
28-107. CRC_REGL3 Register 3405
28-108. CRC_REGH3 Register 3406
28-109. PSA_SECSIGREGL3 Register 3407
28-110. PSA_SECSIGREGH3 Register 3408
28-111. RAW_DATAREGL3 Register 3409
28-112. RAW_DATAREGH3 Register 3410

28-113. CRC_PCOUNT_REG4 Register 3411
28-114. CRC_SCOUNT_REG4 Register 3412
28-115. CRC_CURSEC_REG4 Register 3413
28-116. CRC_WDTOPLD4 Register 3414
28-117. CRC_BCTOPLD4 Register 3415
28-118. PSA_SIGREGL4 Register 3416
28-119. PSA_SIGREGH4 Register 3417
28-120. CRC_REGL4 Register 3418
28-121. CRC_REGH4 Register 3419
28-122. PSA_SECSIGREGL4 Register 3420
28-123. PSA_SECSIGREGH4 Register 3421
28-124. RAW_DATAREGL4 Register 3422
28-125. RAW_DATAREGH4 Register 3423
28-126. MCRC_BUS_SEL Register 3424
28-127. PBIST Block Diagram 3426
28-128. PBIST_DLR Register 3430
28-129. PBIST_PACT Register 3431
28-130. PBIST_ID Register 3432
28-131. PBIST_OVR Register 3433
28-132. PBIST_FSFR0 Register 3434
28-133. PBIST_FSFR1 Register 3435
28-134. PBIST_FSRCR0 Register 3436
28-135. PBIST_FSRCR1 Register 3437
28-136. PBIST_ROM Register 3438
28-137. PBIST_ALGO Register 3439
28-138. PBIST_RINFOL Register 3440
28-139. PBIST_RINFOU Register 3441
28-140. OPMISR Conceptual Diagram 3443
28-141. Block Diagram for STC With Multiple Segments 3444
28-142. STC Flow (1 of 2) 3446
28-143. STC Flow (2 of 2) 3447
28-144. STCGCR0 Register 3456
28-145. STCGCR1 Register 3457
28-146. STCTPR Register 3458
28-147. STC_CADDR Register 3459
28-148. STCCICR Register 3460
28-149. STCGSTAT Register 3461
28-150. STCFSTAT Register 3462
28-151. STCSCSCR Register 3463
28-152. STC_CADDR2 Register 3464
28-153. STC_CLKDIV Register 3465
28-154. STC_SEGPLR Register 3466
28-155. SEG0_START_ADDR Register 3467
28-156. SEG1_START_ADDR Register 3468
28-157. SEG2_START_ADDR Register 3469
28-158. SEG3_START_ADDR Register 3470
28-159. CORE1_CURMISR_0 Register 3471
28-160. CORE1_CURMISR_1 Register 3472
28-161. CORE1_CURMISR_2 Register 3473

28-162. CORE1_CURMISR_3 Register 3474
28-163. CORE1_CURMISR_4 Register 3475
28-164. CORE1_CURMISR_5 Register 3476
28-165. CORE1_CURMISR_6 Register 3477
28-166. CORE1_CURMISR_7 Register 3478
28-167. CORE1_CURMISR_8 Register 3479
28-168. CORE1_CURMISR_9 Register 3480
28-169. CORE1_CURMISR_10 Register 3481
28-170. CORE1_CURMISR_11 Register 3482
28-171. CORE1_CURMISR_12 Register 3483
28-172. CORE1_CURMISR_13 Register 3484
28-173. CORE1_CURMISR_14 Register 3485
28-174. CORE1_CURMISR_15 Register 3486
28-175. CORE1_CURMISR_16 Register 3487
28-176. CORE1_CURMISR_17 Register 3488
28-177. CORE1_CURMISR_18 Register 3489
28-178. CORE1_CURMISR_19 Register 3490
28-179. CORE1_CURMISR_20 Register 3491
28-180. CORE1_CURMISR_21 Register 3492
28-181. CORE1_CURMISR_22 Register 3493
28-182. CORE1_CURMISR_23 Register 3494
28-183. CORE1_CURMISR_24 Register 3495
28-184. CORE1_CURMISR_25 Register 3496
28-185. CORE1_CURMISR_26 Register 3497
28-186. CORE1_CURMISR_27 Register 3498
28-187. CORE2_CURMISR_0 Register 3499
28-188. CORE2_CURMISR_1 Register 3500
28-189. CORE2_CURMISR_2 Register 3501
28-190. CORE2_CURMISR_3 Register 3502
28-191. CORE2_CURMISR_4 Register 3503
28-192. CORE2_CURMISR_5 Register 3504
28-193. CORE2_CURMISR_6 Register 3505
28-194. CORE2_CURMISR_7 Register 3506
28-195. CORE2_CURMISR_8 Register 3507
28-196. CORE2_CURMISR_9 Register 3508
28-197. CORE2_CURMISR_10 Register 3509
28-198. CORE2_CURMISR_11 Register 3510
28-199. CORE2_CURMISR_12 Register 3511
28-200. CORE2_CURMISR_13 Register 3512
28-201. CORE2_CURMISR_14 Register 3513
28-202. CORE2_CURMISR_15 Register 3514
28-203. CORE2_CURMISR_16 Register 3515
28-204. CORE2_CURMISR_17 Register 3516
28-205. CORE2_CURMISR_18 Register 3517
28-206. CORE2_CURMISR_19 Register 3518
28-207. CORE2_CURMISR_20 Register 3519
28-208. CORE2_CURMISR_21 Register 3520
28-209. CORE2_CURMISR_22 Register 3521
28-210. CORE2_CURMISR_23 Register 3522

28-211. CORE2_CURMISR_24 Register 3523
28-212. CORE2_CURMISR_25 Register 3524
28-213. CORE2_CURMISR_26 Register 3525
28-214. CORE2_CURMISR_27 Register 3526
28-215. Core Clock Comparator Module 3526
28-216. Block Diagram 3527

List of Tables

1-1. 14xx Acronyms 136
1-2. Master Subsystem, Cortex-R4F Memory Map 140
1-3. EDMA-TPTC Memory Map 142
1-4. Example Configurations 143
1-5. MSS_CCCA and MSS_CCCB Integration Connections 144
1-6. MSS_DCCA Clock Source Selection Table 145
1-7. MSS_DCCB Clock Source Selection Table 145
1-8. MSS_DMA Request Map 147
1-9. Interrupt Request Assignments 149
1-10. DSS_TPCC Configuration 154
1-11. DSS_TPTC Configuration 155
1-12. EDMA Request Map 156
1-13. MSS_ESM Mapping 157
1-14. 14xx High-Speed Interfaces 160
2-1. 16xx Acronyms 163
2-2. Master Subsystem, Cortex-R4F Memory Map 167
2-3. DSP C674x Memory Map 171
2-4. EDMA-TPTC Memory Map 172
2-5. TCM and Shared Memory Available for Cortex R4F in Master Subsystem 173
2-6. MSS_CCCA and MSS_CCCB Integration Connections 174
2-7. MSS_DCCA Clock Source Selection Table 176
2-8. MSS_DCCB Clock Source Selection Table 176
2-9. DSP Event Assignment 176
2-10. MSS_DMA Request Map 181
2-11. Interrupt Request Assignments 187
2-12. DSS_TPCC Configuration 192
2-13. DSS_TPTC Configuration 193
2-14. EDMA Request Map 195
2-15. MSS_ESM Mapping 199
2-16. DSS_ESM Mapping 202
2-17. 16xx High-Speed Interfaces 203
2-18. 18xx Acronyms 204
2-19. Master Subsystem, Cortex-R4F Memory Map 208
2-20. DSP C674x Memory Map 213
2-21. EDMA-TPTC Memory Map 215
2-22. TCM and Shared Memory Available for Cortex R4F in Master Subsystem 215
2-23. MSS_CCCA and MSS_CCCB Integration Connections 216
2-24. MSS_DCCA Clock Source Selection Table 218
2-25. MSS_DCCB Clock Source Selection Table 218
2-26. DSP Event Assignment 218
2-27. MSS_DMA Request Map 223
2-28. Interrupt Request Assignments 229
2-29. DSS_TPCC Configuration 234
2-30. DSS_TPTC Configuration 235
2-31. EDMA Request Map 237
2-32. MSS_ESM Mapping 241
2-33. DSS_ESM Mapping 244

2-34. 18xx High-Speed Interfaces 245
3-1. Acronyms 248
3-2. Master Subsystem, Cortex-R4F Memory Map 253
3-3. DSP C674x Memory Map 257
3-4. EDMA-TPTC Memory Map 259
3-5. TCM and Shared Memory Available for Cortex R4F in Master Subsystem 260
3-6. MSS_CCCA and MSS_CCCB Integration Connections 261
3-7. MSS_DCCA Clock Source Selection Table 262
3-8. MSS_DCCB Clock Source Selection Table 262
3-9. DSP Event Assignment 263
3-10. MSS_DMA Request Map 267
3-11. Interrupt Request Assignments 273
3-12. DSS_TPCC Configuration 278
3-13. DSS_TPTC Configuration 279
3-14. EDMA Request Map 281
3-15. MSS_ESM Mapping 285
3-16. DSS_ESM Mapping 288
3-17. 68xx/64xx High-Speed Interfaces 289
4-1. PRCM Overview 291
4-2. Clock Sources 292
4-3. Clock Domains 293
4-4. PMIC Registers 294
4-5. Reset Types and Sources 295
4-6. Reset Domains 295
4-7. Reset Cause Registers 296
4-8. Power Domain Status 299
4-9. SOP Table 300
4-10. Pre-Eclipse View of Master Subsystem Cortex-R4F 300
4-11. Post-Eclipse View of Master Subsystem Cortex-R4F 301
4-12. Enable ROM Eclipsing 301
4-13. MSS_TOPRCM Registers 302
4-14. MSS_TOPRCM Access Type Codes 303
4-15. BSSCTL Register Field Descriptions 304
4-16. EXTCLKDIV Register Field Descriptions 305
4-17. EXTCLKSRCSEL Register Field Descriptions 306
4-18. EXTCLKCTL Register Field Descriptions 307
4-19. SOFTSYSRST Register Field Descriptions 308
4-20. WDRSTEN Register Field Descriptions 309
4-21. SYSRSTCAUSE Register Field Descriptions 310
4-22. SYSRSTCAUSECLR Register Field Descriptions 311
4-23. MISCCAPT Register Field Descriptions 312
4-24. DCDCCTL0 Register Field Descriptions 313
4-25. DCDCCTL1 Register Field Descriptions 314
4-26. MISCCTL Register Field Descriptions 315
4-27. USERMODEEN Register Field Descriptions 316
4-28. LVDSPADCTL0 Register Field Descriptions 317
4-29. LVDSPADCTL1 Register Field Descriptions 318
4-30. DFTREG0 Register Field Descriptions 319
4-31. DFTREG1 Register Field Descriptions 320

4-32. DFTREG2 Register Field Descriptions 321
4-33. DFTREG3 Register Field Descriptions 322
4-34. DFTREG4 Register Field Descriptions 323
4-35. DFTREG5 Register Field Descriptions 324
4-36. SHMEMINITADDR Register Field Descriptions 325
4-37. SHMEMINITECC Register Field Descriptions 326
4-38. DSSMEMBANKEN Register Field Descriptions 327
4-39. DSSMEMTAB0 Register Field Descriptions 328
4-40. DSSMEMTAB1 Register Field Descriptions 329
4-41. TCMAMEMBANK_EN Register Field Descriptions 330
4-42. TCMAMEMTAB0 Register Field Descriptions 331
4-43. TCMAMEMTAB1 Register Field Descriptions 332
4-44. TCMBMEMBANKEN Register Field Descriptions 333
4-45. TCMBMEMTAB0 Register Field Descriptions 334
4-46. TCMBMEM_TAB1 Register Field Descriptions 335
4-47. MEMINITSTART Register Field Descriptions 336
4-48. MEMINITDONE Register Field Descriptions 337
4-49. MSS_SIGNATURE Register Field Descriptions 338
4-50. MISCCTL1 Register Field Descriptions 339
4-51. USERMODEEN2 Register Field Descriptions 340
4-52. SYSTICK Register Field Descriptions 341
4-53. MSS_RCM Registers 342
4-54. MSS_RCM Access Type Codes 342
4-55. SOFTRST2 Register Field Descriptions 344
4-56. CLKDIVCTL0 Register Field Descriptions 345
4-57. CLKSRCSEL0 Register Field Descriptions 346
4-58. CR4CTL Register Field Descriptions 347
4-59. CLKGATE Register Field Descriptions 348
4-60. CLKSRCSEL1 Register Field Descriptions 349
4-61. CURRCLKDIV0 Register Field Descriptions 350
4-62. MEMINITSTART Register Field Descriptions 351
4-63. CURRCLKDIV1 Register Field Descriptions 352
4-64. MEMINITDONE Register Field Descriptions 353
4-65. USERMODEEN Register Field Descriptions 354
4-66. NSYSPERUSERMODEN Register Field Descriptions 355
4-67. ESMGATE0 Register Field Descriptions 356
4-68. ESMGATE1 Register Field Descriptions 357
4-69. ESMGATE2 Register Field Descriptions 358
4-70. ESMGATE3 Register Field Descriptions 359
4-71. ESMGATE4 Register Field Descriptions 360
4-72. KEY Register Field Descriptions 361
4-73. SWIRQA Register Field Descriptions 362
4-74. SWIRQB Register Field Descriptions 363
4-75. MISCCTL0 Register Field Descriptions 364
4-76. ATCMERRCAPTCTL Register Field Descriptions 365
4-77. B0TCMERRCAPTCTL Register Field Descriptions 366
4-78. B1TCMERRCAPTCTL Register Field Descriptions 367
4-79. SOFTCORERST Register Field Descriptions 368
4-80. RSTCAUSE Register Field Descriptions 369

4-81. RSTCAUSECLR Register Field Descriptions 370
4-82. SPITRIGSRC Register Field Descriptions 371
4-83. CLKINUSE Register Field Descriptions 372
4-84. ECCEN Register Field Descriptions 373
4-85. ECCCAPT Register Field Descriptions 374
4-86. CLKDIVCTL2 Register Field Descriptions 375
4-87. SWIRQC Register Field Descriptions 376
4-88. MSS_GPCFG_REG Registers 377
4-89. MSS_GPCFG_REG Access Type Codes 377
4-90. GPCFG0 Register Field Descriptions 378
4-91. GPCFG1 Register Field Descriptions 379
4-92. GPCFG2 Register Field Descriptions 380
4-93. GPCFG3 Register Field Descriptions 381
4-94. GPCFG4 Register Field Descriptions 382
4-95. CCCACFG0 Register Field Descriptions 383
4-96. CCCACFG1 Register Field Descriptions 384
4-97. CCCACFG2 Register Field Descriptions 385
4-98. CCCACFG3 Register Field Descriptions 386
4-99. CCCBCFG0 Register Field Descriptions 387
4-100. CCCBCFG1 Register Field Descriptions 388
4-101. CCCBCFG2 Register Field Descriptions 389
4-102. CCCBCFG3 Register Field Descriptions 390
4-103. CCCACNTVAL Register Field Descriptions 391
4-104. CCCBCNTVAL Register Field Descriptions 392
4-105. CCCABERRSTAT Register Field Descriptions 393
4-106. USERMODEEN Register Field Descriptions 394
4-107. DSS_REG Registers 395
4-108. DSS_REG Access Type Codes 397
4-109. RTIEVENTCAPTURESEL Register Field Descriptions 398
4-110. ADCBUFCFG1 Register Field Descriptions 399
4-111. ADCBUFCFG2 Register Field Descriptions 400
4-112. ADCBUFCFG3 Register Field Descriptions 401
4-113. ADCBUFCFG4 Register Field Descriptions 402
4-114. CQCFG1 Register Field Descriptions 403
4-115. TPCCPARSTATCFG Register Field Descriptions 404
4-116. CSI2TXPARSTATCFG Register Field Descriptions 405
4-117. CSICFG1 Register Field Descriptions 406
4-118. TPTC0WRMPUSTADD0 Register Field Descriptions 407
4-119. TPTC0WRMPUSTADD1 Register Field Descriptions 408
4-120. TPTC0WRMPUSTADD2 Register Field Descriptions 409
4-121. TPTC0WRMPUSTADD3 Register Field Descriptions 410
4-122. TPTC0WRMPUSTADD4 Register Field Descriptions 411
4-123. TPTC0WRMPUSTADD5 Register Field Descriptions 412
4-124. TPTC0WRMPUSTADD6 Register Field Descriptions 413
4-125. TPTC0WRMPUSTADD7 Register Field Descriptions 414
4-126. TPTC0WRMPUENDADD0 Register Field Descriptions 415
4-127. TPTC0WRMPUENDADD1 Register Field Descriptions 416
4-128. TPTC0WRMPUENDADD2 Register Field Descriptions 417
4-129. TPTC0WRMPUENDADD3 Register Field Descriptions 418

4-130. TPTC0WRMPUENDADD4 Register Field Descriptions 419
4-131. TPTC0WRMPUENDADD5 Register Field Descriptions 420
4-132. TPTC0WRMPUENDADD6 Register Field Descriptions 421
4-133. TPTC0WRMPUENDADD7 Register Field Descriptions 422
4-134. TPTC0WRMPUERRADD Register Field Descriptions 423
4-135. TPTC0RDMPUSTADD0 Register Field Descriptions 424
4-136. TPTC0RDMPUSTADD1 Register Field Descriptions 425
4-137. TPTC0RDMPUSTADD2 Register Field Descriptions 426
4-138. TPTC0RDMPUSTADD3 Register Field Descriptions 427
4-139. TPTC0RDMPUSTADD4 Register Field Descriptions 428
4-140. TPTC0RDMPUSTADD5 Register Field Descriptions 429
4-141. TPTC0RDMPUSTADD6 Register Field Descriptions 430
4-142. TPTC0RDMPUSTADD7 Register Field Descriptions 431
4-143. TPTC0RDMPUENDADD0 Register Field Descriptions 432
4-144. TPTC0RDMPUENDADD1 Register Field Descriptions 433
4-145. TPTC0RDMPUENDADD2 Register Field Descriptions 434
4-146. TPTC0RDMPUENDADD3 Register Field Descriptions 435
4-147. TPTC0RDMPUENDADD4 Register Field Descriptions 436
4-148. TPTC0RDMPUENDADD5 Register Field Descriptions 437
4-149. TPTC0RDMPUENDADD6 Register Field Descriptions 438
4-150. TPTC0RDMPUENDADD7 Register Field Descriptions 439
4-151. TPTC0RDMPUERRADD Register Field Descriptions 440
4-152. TPTC1WRMPUSTADD0 Register Field Descriptions 441
4-153. TPTC1WRMPUSTADD1 Register Field Descriptions 442
4-154. TPTC1WRMPUSTADD2 Register Field Descriptions 443
4-155. TPTC1WRMPUSTADD3 Register Field Descriptions 444
4-156. TPTC1WRMPUSTADD4 Register Field Descriptions 445
4-157. TPTC1WRMPUSTADD5 Register Field Descriptions 446
4-158. TPTC1WRMPUSTADD6 Register Field Descriptions 447
4-159. TPTC1WRMPUSTADD7 Register Field Descriptions 448
4-160. TPTC1WRMPUENDADD0 Register Field Descriptions 449
4-161. TPTC1WRMPUENDADD1 Register Field Descriptions 450
4-162. TPTC1WRMPUENDADD2 Register Field Descriptions 451
4-163. TPTC1WRMPUENDADD3 Register Field Descriptions 452
4-164. TPTC1WRMPUENDADD4 Register Field Descriptions 453
4-165. TPTC1WRMPUENDADD5 Register Field Descriptions 454
4-166. TPTC1WRMPUENDADD6 Register Field Descriptions 455
4-167. TPTC1WRMPUENDADD7 Register Field Descriptions 456
4-168. TPTC1WRMPUERRADD Register Field Descriptions 457
4-169. TPTC1RDMPUSTADD0 Register Field Descriptions 458
4-170. TPTC1RDMPUSTADD1 Register Field Descriptions 459
4-171. TPTC1RDMPUSTADD2 Register Field Descriptions 460
4-172. TPTC1RDMPUSTADD3 Register Field Descriptions 461
4-173. TPTC1RDMPUSTADD4 Register Field Descriptions 462
4-174. TPTC1RDMPUSTADD5 Register Field Descriptions 463
4-175. TPTC1RDMPUSTADD6 Register Field Descriptions 464
4-176. TPTC1RDMPUSTADD7 Register Field Descriptions 465
4-177. TPTC1RDMPUENDADD0 Register Field Descriptions 466
4-178. TPTC1RDMPUENDADD1 Register Field Descriptions 467

4-179. TPTC1RDMPUENDADD2 Register Field Descriptions 468
4-180. TPTC1RDMPUENDADD3 Register Field Descriptions 469
4-181. TPTC1RDMPUENDADD4 Register Field Descriptions 470
4-182. TPTC1RDMPUENDADD5 Register Field Descriptions 471
4-183. TPTC1RDMPUENDADD6 Register Field Descriptions 472
4-184. TPTC1RDMPUENDADD7 Register Field Descriptions 473
4-185. TPTC1RDMPUERRADD Register Field Descriptions 474
4-186. TPTCMPUVALIDCFG Register Field Descriptions 475
4-187. TPTCMPUENCFG Register Field Descriptions 476
4-188. TESTPATTERNRX1ICFG Register Field Descriptions 477
4-189. TESTPATTERNRX2ICFG Register Field Descriptions 478
4-190. TESTPATTERNRX3ICFG Register Field Descriptions 479
4-191. TESTPATTERNRX4ICFG Register Field Descriptions 480
4-192. TESTPATTERNRX1QCFG Register Field Descriptions 481
4-193. TESTPATTERNRX2QCFG Register Field Descriptions 482
4-194. TESTPATTERNRX3QCFG Register Field Descriptions 483
4-195. TESTPATTERNRX4QCFG Register Field Descriptions 484
4-196. TESTPATTERNVLDCFG Register Field Descriptions 485
4-197. DSSMISC Register Field Descriptions 486
4-198. DSSMISC2 Register Field Descriptions 487
4-199. MSS_TOPRCM Registers 488
4-200. MSS_TOPRCM Access Type Codes 489
4-201. BSSCTL Register Field Descriptions 490
4-202. DSSCTL Register Field Descriptions 491
4-203. EXTCLKDIV Register Field Descriptions 492
4-204. EXTCLKSRCSEL Register Field Descriptions 493
4-205. EXTCLKCTL Register Field Descriptions 494
4-206. SOFTSYSRST Register Field Descriptions 495
4-207. WDRSTEN Register Field Descriptions 496
4-208. SYSRSTCAUSE Register Field Descriptions 497
4-209. SYSRSTCAUSECLR Register Field Descriptions 498
4-210. MISCCAPT Register Field Descriptions 499
4-211. DCDCCTL0 Register Field Descriptions 500
4-212. DCDCCTL1 Register Field Descriptions 501
4-213. USERMODEEN Register Field Descriptions 502
4-214. LVDSPADCTL0 Register Field Descriptions 503
4-215. LVDSPADCTL1 Register Field Descriptions 504
4-216. DFTREG0 Register Field Descriptions 505
4-217. DFTREG1 Register Field Descriptions 506
4-218. DFTREG5 Register Field Descriptions 507
4-219. MEMINITDONE Register Field Descriptions 508
4-220. SPARE0_3 Register Field Descriptions 509
4-221. MSS_SIGNATURE Register Field Descriptions 510
4-222. GEMBOOTSTCEN Register Field Descriptions 511
4-223. MISCCTL1 Register Field Descriptions 512
4-224. USERMODEEN2 Register Field Descriptions 513
4-225. SYSTICK Register Field Descriptions 514
4-226. SECURECFGREG1 Register Field Descriptions 515
4-227. SECURECFGREG2 Register Field Descriptions 516

4-228. SECURECFGREG3 Register Field Descriptions 517
4-229. SECURECFGREG4 Register Field Descriptions 518
4-230. SECURERAMREG Register Field Descriptions 519
4-231. SPAREMULTIBIT Register Field Descriptions 520
4-232. SPARE4_9 Register Field Descriptions 521
4-233. UID31TO0 Register Field Descriptions 522
4-234. UID63TO32 Register Field Descriptions 523
4-235. UID95TO64 Register Field Descriptions 524
4-236. UID119TO96 Register Field Descriptions 525
4-237. MEMINITSTARTSHMEM Register Field Descriptions 526
4-238. MEMINITDONESHMEM Register Field Descriptions 527
4-239. DSSMEMTAB0 Register Field Descriptions 528
4-240. TCMAMEMTAB Register Field Descriptions 529
4-241. TCMBMEMTAB Register Field Descriptions 530
4-242. SHMEMBANKSEL3TO0 Register Field Descriptions 531
4-243. SHMEMBANKSEL7TO4 Register Field Descriptions 532
4-244. PBISTCLKCTL Register Field Descriptions 533
4-245. MSS_RCM Registers 534
4-246. MSS_RCM Access Type Codes 535
4-247. SOFTRST1 Register Field Descriptions 536
4-248. SOFTRST2 Register Field Descriptions 537
4-249. CLKDIVCTL0 Register Field Descriptions 538
4-250. CLKSRCSEL0 Register Field Descriptions 539
4-251. CR4CTL Register Field Descriptions 540
4-252. CLKGATE Register Field Descriptions 541
4-253. CLKSRCSEL1 Register Field Descriptions 542
4-254. CURRCLKDIV0 Register Field Descriptions 543
4-255. RTICURRCLKDIV Register Field Descriptions 544
4-256. MEMINITSTART Register Field Descriptions 545
4-257. CURRCLKDIV1 Register Field Descriptions 546
4-258. MEMINITDONE Register Field Descriptions 547
4-259. ECCENMSSGEM Register Field Descriptions 548
4-260. ECCCAPTMSSGEM Register Field Descriptions 549
4-261. ECCENBSSGEM Register Field Descriptions 550
4-262. ECCCAPTBSSGEM Register Field Descriptions 551
4-263. USERMODEEN Register Field Descriptions 552
4-264. NSYSPERUSERMODEN Register Field Descriptions 553
4-265. SECURERAMMMI Register Field Descriptions 554
4-266. SECURERAMECC Register Field Descriptions 555
4-267. ESMGATE0 Register Field Descriptions 556
4-268. ESMGATE1 Register Field Descriptions 557
4-269. ESMGATE2 Register Field Descriptions 558
4-270. ESMGATE3 Register Field Descriptions 559
4-271. ESMGATE4 Register Field Descriptions 560
4-272. KEY Register Field Descriptions 561
4-273. SWIRQA Register Field Descriptions 562
4-274. SWIRQB Register Field Descriptions 563
4-275. MISCCTL0 Register Field Descriptions 564
4-276. ATCMERRCAPTCTL Register Field Descriptions 565

4-277. B0TCMERRCAPTCTL Register Field Descriptions 566
4-278. B1TCMERRCAPTCTL Register Field Descriptions 567
4-279. SOFTCORERST Register Field Descriptions 568
4-280. RSTCAUSE Register Field Descriptions 569
4-281. RSTCAUSECLR Register Field Descriptions 570
4-282. SPITRIGSRC Register Field Descriptions 571
4-283. CLKINUSE Register Field Descriptions 572
4-284. ECCENMSSBSS Register Field Descriptions 573
4-285. ECCCAPTMSSBSS Register Field Descriptions 574
4-286. CLKDIVCTL2 Register Field Descriptions 575
4-287. SWIRQC Register Field Descriptions 576
4-288. MSS_GPCFG_REG Registers 577
4-289. MSS_GPCFG_REG Access Type Codes 577
4-290. GPCFG0 Register Field Descriptions 579
4-291. GPCFG1 Register Field Descriptions 580
4-292. GPCFG2 Register Field Descriptions 581
4-293. GPCFG3 Register Field Descriptions 582
4-294. GPCFG4 Register Field Descriptions 583
4-295. GPCFG6 Register Field Descriptions 584
4-296. GPCFG11 Register Field Descriptions 585
4-297. CCCACFG0 Register Field Descriptions 586
4-298. CCCACFG1 Register Field Descriptions 587
4-299. CCCACFG2 Register Field Descriptions 588
4-300. CCCACFG3 Register Field Descriptions 589
4-301. CCCBCFG0 Register Field Descriptions 590
4-302. CCCBCFG1 Register Field Descriptions 591
4-303. CCCBCFG2 Register Field Descriptions 592
4-304. CCCBCFG3 Register Field Descriptions 593
4-305. CCCACNTVAL Register Field Descriptions 594
4-306. CCCBCNTVAL Register Field Descriptions 595
4-307. CCCABERRSTAT Register Field Descriptions 596
4-308. USERMODEEN Register Field Descriptions 597
4-309. EPWMCFG Register Field Descriptions 598
4-310. DMMSWINT0 Register Field Descriptions 599
4-311. DMMSWINT1 Register Field Descriptions 600
4-312. DMMSWINTSEL0 Register Field Descriptions 601
4-313. DMMSWINTSEL1 Register Field Descriptions 602
4-314. CCCBWDEN Register Field Descriptions 603
4-315. GPIOINTREDGESEL Register Field Descriptions 604
4-316. PWMDMATRIGEN Register Field Descriptions 605
4-317. JTAGTXDATA Register Field Descriptions 606
4-318. JTAGTXCONTROL Register Field Descriptions 607
4-319. JTAGRXDATA Register Field Descriptions 608
4-320. JTAGTXRXACK Register Field Descriptions 609
4-321. JTAGRXCONTROL Register Field Descriptions 610
4-322. MSS2GEMSWIRQ Register Field Descriptions 611
4-323. CSETBFLUSH Register Field Descriptions 612
4-324. DSS_REG Registers 613
4-325. DSS_REG Access Type Codes 615

4-326. RTIEVENTCAPTURESEL Register Field Descriptions 616
4-327. CQCFG1 Register Field Descriptions 617
4-328. TPCCPARSTATCFG Register Field Descriptions 618
4-329. TPTC0WRMPUSTADD0 Register Field Descriptions 619
4-330. TPTC0WRMPUSTADD1 Register Field Descriptions 620
4-331. TPTC0WRMPUSTADD2 Register Field Descriptions 621
4-332. TPTC0WRMPUSTADD3 Register Field Descriptions 622
4-333. TPTC0WRMPUSTADD4 Register Field Descriptions 623
4-334. TPTC0WRMPUSTADD5 Register Field Descriptions 624
4-335. TPTC0WRMPUENDADD0 Register Field Descriptions 625
4-336. TPTC0WRMPUENDADD1 Register Field Descriptions 626
4-337. TPTC0WRMPUENDADD2 Register Field Descriptions 627
4-338. TPTC0WRMPUENDADD3 Register Field Descriptions 628
4-339. TPTC0WRMPUENDADD4 Register Field Descriptions 629
4-340. TPTC0WRMPUENDADD5 Register Field Descriptions 630
4-341. TPTC0WRMPUERRADD Register Field Descriptions 631
4-342. TPTC0RDMPUSTADD0 Register Field Descriptions 632
4-343. TPTC0RDMPUSTADD1 Register Field Descriptions 633
4-344. TPTC0RDMPUSTADD2 Register Field Descriptions 634
4-345. TPTC0RDMPUSTADD3 Register Field Descriptions 635
4-346. TPTC0RDMPUSTADD4 Register Field Descriptions 636
4-347. TPTC0RDMPUSTADD5 Register Field Descriptions 637
4-348. TPTC0RDMPUENDADD0 Register Field Descriptions 638
4-349. TPTC0RDMPUENDADD1 Register Field Descriptions 639
4-350. TPTC0RDMPUENDADD2 Register Field Descriptions 640
4-351. TPTC0RDMPUENDADD3 Register Field Descriptions 641
4-352. TPTC0RDMPUENDADD4 Register Field Descriptions 642
4-353. TPTC0RDMPUENDADD5 Register Field Descriptions 643
4-354. TPTC0RDMPUERRADD Register Field Descriptions 644
4-355. TPTC1WRMPUSTADD0 Register Field Descriptions 645
4-356. TPTC1WRMPUSTADD1 Register Field Descriptions 646
4-357. TPTC1WRMPUSTADD2 Register Field Descriptions 647
4-358. TPTC1WRMPUSTADD3 Register Field Descriptions 648
4-359. TPTC1WRMPUSTADD4 Register Field Descriptions 649
4-360. TPTC1WRMPUSTADD5 Register Field Descriptions 650
4-361. TPTC1WRMPUENDADD0 Register Field Descriptions 651
4-362. TPTC1WRMPUENDADD1 Register Field Descriptions 652
4-363. TPTC1WRMPUENDADD2 Register Field Descriptions 653
4-364. TPTC1WRMPUENDADD3 Register Field Descriptions 654
4-365. TPTC1WRMPUENDADD4 Register Field Descriptions 655
4-366. TPTC1WRMPUENDADD5 Register Field Descriptions 656
4-367. TPTC1WRMPUERRADD Register Field Descriptions 657
4-368. TPTC1RDMPUSTADD0 Register Field Descriptions 658
4-369. TPTC1RDMPUSTADD1 Register Field Descriptions 659
4-370. TPTC1RDMPUSTADD2 Register Field Descriptions 660
4-371. TPTC1RDMPUSTADD3 Register Field Descriptions 661
4-372. TPTC1RDMPUSTADD4 Register Field Descriptions 662
4-373. TPTC1RDMPUSTADD5 Register Field Descriptions 663
4-374. TPTC1RDMPUENDADD0 Register Field Descriptions 664

4-375. TPTC1RDMPUENDADD1 Register Field Descriptions 665
4-376. TPTC1RDMPUENDADD2 Register Field Descriptions 666
4-377. TPTC1RDMPUENDADD3 Register Field Descriptions 667
4-378. TPTC1RDMPUENDADD4 Register Field Descriptions 668
4-379. TPTC1RDMPUENDADD5 Register Field Descriptions 669
4-380. TPTC1RDMPUERRADD Register Field Descriptions 670
4-381. TPTCMPUVALIDCFG Register Field Descriptions 671
4-382. TPTCMPUENCFG Register Field Descriptions 672
4-383. TESTPATTERNRX1ICFG Register Field Descriptions 673
4-384. TESTPATTERNRX2ICFG Register Field Descriptions 674
4-385. TESTPATTERNRX3ICFG Register Field Descriptions 675
4-386. TESTPATTERNRX4ICFG Register Field Descriptions 676
4-387. TESTPATTERNRX1QCFG Register Field Descriptions 677
4-388. TESTPATTERNRX2QCFG Register Field Descriptions 678
4-389. TESTPATTERNRX3QCFG Register Field Descriptions 679
4-390. TESTPATTERNRX4QCFG Register Field Descriptions 680
4-391. TESTPATTERNVLDCFG Register Field Descriptions 681
4-392. TPCC1PARSTATCFG Register Field Descriptions 682
4-393. DMMSWINT1 Register Field Descriptions 683
4-394. DSSINTRCFG Register Field Descriptions 684
4-395. MPUMSTIDCFG1 Register Field Descriptions 685
4-396. MPUMSTIDCFG2 Register Field Descriptions 686
4-397. MPUMSTIDCFG3 Register Field Descriptions 687
4-398. HSRAM1ECCCFG Register Field Descriptions 688
4-399. DATATRRAMECCCFG Register Field Descriptions 689
4-400. ADCBUFPINGECCCFG Register Field Descriptions 690
4-401. ADCBUFPONGECCCFG Register Field Descriptions 691
4-402. UMAP0PARITYCFG1 Register Field Descriptions 692
4-403. UMAP0PARITYCFG2 Register Field Descriptions 693
4-404. UMAP0PARITYCFG3 Register Field Descriptions 694
4-405. UMAP1PARITYCFG1 Register Field Descriptions 695
4-406. UMAP1PARITYCFG2 Register Field Descriptions 696
4-407. UMAP1PARITYCFG3 Register Field Descriptions 697
4-408. ESMGRP2MASKCFG Register Field Descriptions 698
4-409. L2MEMINITCFG1 Register Field Descriptions 699
4-410. L2MEMINITCFG2 Register Field Descriptions 701
4-411. GEMRSTCAUSE Register Field Descriptions 702
4-412. GEMPWRSMCFG4 Register Field Descriptions 703
4-413. PWRSMWAKEMASK0 Register Field Descriptions 704
4-414. PWRSMWAKEMASK1 Register Field Descriptions 705
4-415. PWRSMWAKEMASK2 Register Field Descriptions 706
4-416. PWRSMMISEVTMASK0 Register Field Descriptions 707
4-417. PWRSMMISEVTMASK1 Register Field Descriptions 708
4-418. PWRSMMISEVTMASK2 Register Field Descriptions 709
4-419. PWRSMWAKESRCSTAT0 Register Field Descriptions 710
4-420. PWRSMWAKESRCSTAT1 Register Field Descriptions 711
4-421. PWRSMWAKESRCSTAT2 Register Field Descriptions 712
4-422. PWRSMEVNTMONSTAT0 Register Field Descriptions 713
4-423. PWRSMEVNTMONSTAT1 Register Field Descriptions 714

4-424. PWRSMEVNTMONSTAT2 Register Field Descriptions 715
4-425. PWRSMWAKESRCSTATCLR0 Register Field Descriptions 716
4-426. PWRSMWAKESRCSTATCLR1 Register Field Descriptions 717
4-427. PWRSMWAKESRCSTATCLR2 Register Field Descriptions 718
4-428. ADCBUFCFG1 Register Field Descriptions 719
4-429. ADCBUFCFG2 Register Field Descriptions 720
4-430. ADCBUFCFG3 Register Field Descriptions 721
4-431. ADCBUFCFG4 Register Field Descriptions 722
4-432. STCPBISTSMCFG1 Register Field Descriptions 723
4-433. STCPBISTSMCFG2 Register Field Descriptions 724
4-434. RTI2EVENTCAPTURESEL Register Field Descriptions 725
4-435. DSSMISC5 Register Field Descriptions 726
4-436. DSS_REG2 Registers 727
4-437. DSS_REG2 Access Type Codes 728
4-438. TPTC2WRMPUSTADD0 Register Field Descriptions 729
4-439. TPTC2WRMPUSTADD1 Register Field Descriptions 730
4-440. TPTC2WRMPUSTADD2 Register Field Descriptions 731
4-441. TPTC2WRMPUSTADD3 Register Field Descriptions 732
4-442. TPTC2WRMPUSTADD4 Register Field Descriptions 733
4-443. TPTC2WRMPUSTADD5 Register Field Descriptions 734
4-444. TPTC2WRMPUENDADD0 Register Field Descriptions 735
4-445. TPTC2WRMPUENDADD1 Register Field Descriptions 736
4-446. TPTC2WRMPUENDADD2 Register Field Descriptions 737
4-447. TPTC2WRMPUENDADD3 Register Field Descriptions 738
4-448. TPTC2WRMPUENDADD4 Register Field Descriptions 739
4-449. TPTC2WRMPUENDADD5 Register Field Descriptions 740
4-450. TPTC2WRMPUERRADD Register Field Descriptions 741
4-451. TPTC2RDMPUSTADD0 Register Field Descriptions 742
4-452. TPTC2RDMPUSTADD1 Register Field Descriptions 743
4-453. TPTC2RDMPUSTADD2 Register Field Descriptions 744
4-454. TPTC2RDMPUSTADD3 Register Field Descriptions 745
4-455. TPTC2RDMPUSTADD4 Register Field Descriptions 746
4-456. TPTC2RDMPUSTADD5 Register Field Descriptions 747
4-457. TPTC2RDMPUENDADD0 Register Field Descriptions 748
4-458. TPTC2RDMPUENDADD1 Register Field Descriptions 749
4-459. TPTC2RDMPUENDADD2 Register Field Descriptions 750
4-460. TPTC2RDMPUENDADD3 Register Field Descriptions 751
4-461. TPTC2RDMPUENDADD4 Register Field Descriptions 752
4-462. TPTC2RDMPUENDADD5 Register Field Descriptions 753
4-463. TPTC2RDMPUERRADD Register Field Descriptions 754
4-464. TPTC3WRMPUSTADD0 Register Field Descriptions 755
4-465. TPTC3WRMPUSTADD1 Register Field Descriptions 756
4-466. TPTC3WRMPUSTADD2 Register Field Descriptions 757
4-467. TPTC3WRMPUSTADD3 Register Field Descriptions 758
4-468. TPTC3WRMPUSTADD4 Register Field Descriptions 759
4-469. TPTC3WRMPUSTADD5 Register Field Descriptions 760
4-470. TPTC3WRMPUENDADD0 Register Field Descriptions 761
4-471. TPTC3WRMPUENDADD1 Register Field Descriptions 762
4-472. TPTC3WRMPUENDADD2 Register Field Descriptions 763

4-473. TPTC3WRMPUENDADD3 Register Field Descriptions 764
4-474. TPTC3WRMPUENDADD4 Register Field Descriptions 765
4-475. TPTC3WRMPUENDADD5 Register Field Descriptions 766
4-476. TPTC3WRMPUERRADD Register Field Descriptions 767
4-477. TPTC3RDMPUSTADD0 Register Field Descriptions 768
4-478. TPTC3RDMPUSTADD1 Register Field Descriptions 769
4-479. TPTC3RDMPUSTADD2 Register Field Descriptions 770
4-480. TPTC3RDMPUSTADD3 Register Field Descriptions 771
4-481. TPTC3RDMPUSTADD4 Register Field Descriptions 772
4-482. TPTC3RDMPUSTADD5 Register Field Descriptions 773
4-483. TPTC3RDMPUENDADD0 Register Field Descriptions 774
4-484. TPTC3RDMPUENDADD1 Register Field Descriptions 775
4-485. TPTC3RDMPUENDADD2 Register Field Descriptions 776
4-486. TPTC3RDMPUENDADD3 Register Field Descriptions 777
4-487. TPTC3RDMPUENDADD4 Register Field Descriptions 778
4-488. TPTC3RDMPUENDADD5 Register Field Descriptions 779
4-489. TPTC3RDMPUERRADD Register Field Descriptions 780
4-490. TPTCMPUVALIDCFG2 Register Field Descriptions 781
4-491. TPTCMPUENCFG2 Register Field Descriptions 782
4-492. L3ECCCFG1 Register Field Descriptions 783
4-493. L3ECCCFG2 Register Field Descriptions 784
4-494. DSS2MSSSWIRQ Register Field Descriptions 785
4-495. MSS_TOPRCM Registers 787
4-496. MSS_TOPRCM Access Type Codes 788
4-497. BSSCTL Register Field Descriptions 789
4-498. DSSCTL Register Field Descriptions 790
4-499. EXTCLKDIV Register Field Descriptions 791
4-500. EXTCLKSRCSEL Register Field Descriptions 792
4-501. EXTCLKCTL Register Field Descriptions 793
4-502. SOFTSYSRST Register Field Descriptions 794
4-503. WDRSTEN Register Field Descriptions 795
4-504. SYSRSTCAUSE Register Field Descriptions 796
4-505. SYSRSTCAUSECLR Register Field Descriptions 797
4-506. MISCCAPT Register Field Descriptions 798
4-507. DCDCCTL0 Register Field Descriptions 799
4-508. DCDCCTL1 Register Field Descriptions 800
4-509. USERMODEEN Register Field Descriptions 801
4-510. LVDSPADCTL0 Register Field Descriptions 802
4-511. LVDSPADCTL1 Register Field Descriptions 803
4-512. DFTREG0 Register Field Descriptions 804
4-513. DFTREG1 Register Field Descriptions 805
4-514. DFTREG5 Register Field Descriptions 806
4-515. MEMINITDONE Register Field Descriptions 807
4-516. SPARE0_3 Register Field Descriptions 808
4-517. MSS_SIGNATURE Register Field Descriptions 809
4-518. GEMBOOTSTCEN Register Field Descriptions 810
4-519. MISCCTL1 Register Field Descriptions 811
4-520. USERMODEEN2 Register Field Descriptions 812
4-521. SYSTICK Register Field Descriptions 813

4-522. SECURECFGREG1 Register Field Descriptions 814
4-523. SECURECFGREG2 Register Field Descriptions 815
4-524. SECURECFGREG3 Register Field Descriptions 816
4-525. SECURECFGREG4 Register Field Descriptions 817
4-526. SECURERAMREG Register Field Descriptions 818
4-527. SPAREMULTIBIT Register Field Descriptions 819
4-528. SPARE4_9 Register Field Descriptions 820
4-529. UID31TO0 Register Field Descriptions 821
4-530. UID63TO32 Register Field Descriptions 822
4-531. UID95TO64 Register Field Descriptions 823
4-532. UID119TO96 Register Field Descriptions 824
4-533. MEMINITSTARTSHMEM Register Field Descriptions 825
4-534. MEMINITDONESHMEM Register Field Descriptions 826
4-535. DSSMEMTAB0 Register Field Descriptions 827
4-536. TCMAMEMTAB Register Field Descriptions 828
4-537. TCMBMEMTAB Register Field Descriptions 829
4-538. SHMEMBANKSEL3TO0 Register Field Descriptions 830
4-539. SHMEMBANKSEL7TO4 Register Field Descriptions 831
4-540. PBISTCLKCTL Register Field Descriptions 832
4-541. MSS_RCM Registers 833
4-542. SOFTRST0 Register Field Descriptions 835
4-543. SOFTRST1 Register Field Descriptions 836
4-544. SOFTRST2 Register Field Descriptions 837
4-545. SOFTRST3 Register Field Descriptions 838
4-546. MCUIFSOFTRST0 Register Field Descriptions 839
4-547. ECUIFSOFTRST0 Register Field Descriptions 840
4-548. CLKDIVCTL0 Register Field Descriptions 841
4-549. CLKSRCSEL0 Register Field Descriptions 842
4-550. CR4CTL Register Field Descriptions 843
4-551. SOFTRST4 Register Field Descriptions 844
4-552. CLKGATE Register Field Descriptions 845
4-553. CLKSRCSEL1 Register Field Descriptions 846
4-554. CLKDIVCTL1 Register Field Descriptions 847
4-555. CURRCLKDIV0 Register Field Descriptions 848
4-556. RTICURRCLKDIV Register Field Descriptions 849
4-557. MEMINITSTART Register Field Descriptions 850
4-558. CURRCLKDIV1 Register Field Descriptions 851
4-559. MEMINITDONE Register Field Descriptions 852
4-560. ECCENMSSGEM Register Field Descriptions 853
4-561. ECCCAPTMSSGEM Register Field Descriptions 854
4-562. ECCENBSSGEM Register Field Descriptions 855
4-563. ECCCAPTBSSGEM Register Field Descriptions 856
4-564. USERMODEEN Register Field Descriptions 857
4-565. NSYSPERUSERMODEN Register Field Descriptions 858
4-566. SECURERAMMMI Register Field Descriptions 859
4-567. SECURERAMECC Register Field Descriptions 860
4-568. ESMGATE0 Register Field Descriptions 861
4-569. ESMGATE1 Register Field Descriptions 862
4-570. ESMGATE2 Register Field Descriptions 863

4-571. ESMGATE3 Register Field Descriptions 864
4-572. ESMGATE4 Register Field Descriptions 865
4-573. MSSECOSPARE Register Field Descriptions 866
4-574. KEY Register Field Descriptions 867
4-575. DBGACKCTL0 Register Field Descriptions 868
4-576. DBGACKCTL1 Register Field Descriptions 869
4-577. SWIRQA Register Field Descriptions 870
4-578. SWIRQB Register Field Descriptions 871
4-579. MISCCTL0 Register Field Descriptions 872
4-580. ATCMERRCAPTCTL Register Field Descriptions 873
4-581. B0TCMERRCAPTCTL Register Field Descriptions 874
4-582. B1TCMERRCAPTCTL Register Field Descriptions 875
4-583. SOFTCORERST Register Field Descriptions 876
4-584. WDOGRSTEN Register Field Descriptions 877
4-585. RSTCAUSE Register Field Descriptions 878
4-586. RSTCAUSECLR Register Field Descriptions 879
4-587. SPITRIGSRC Register Field Descriptions 880
4-588. CLKINUSE Register Field Descriptions 881
4-589. ECCENMSSBSS Register Field Descriptions 882
4-590. ECCCAPTMSSBSS Register Field Descriptions 883
4-591. CLKDIVCTL2 Register Field Descriptions 884
4-592. QSPIMCONNECT Register Field Descriptions 885
4-593. QSPISCONNECT Register Field Descriptions 886
4-594. SWIRQC Register Field Descriptions 887
4-595. MSS_GPCFG_REG Registers 888
4-596. MSS_GPCFG_REG Access Type Codes 888
4-597. GPCFG0 Register Field Descriptions 890
4-598. GPCFG1 Register Field Descriptions 891
4-599. GPCFG2 Register Field Descriptions 892
4-600. GPCFG3 Register Field Descriptions 893
4-601. GPCFG4 Register Field Descriptions 894
4-602. GPCFG6 Register Field Descriptions 895
4-603. GPCFG11 Register Field Descriptions 896
4-604. CCCACFG0 Register Field Descriptions 897
4-605. CCCACFG1 Register Field Descriptions 898
4-606. CCCACFG2 Register Field Descriptions 899
4-607. CCCACFG3 Register Field Descriptions 900
4-608. CCCBCFG0 Register Field Descriptions 901
4-609. CCCBCFG1 Register Field Descriptions 902
4-610. CCCBCFG2 Register Field Descriptions 903
4-611. CCCBCFG3 Register Field Descriptions 904
4-612. CCCACNTVAL Register Field Descriptions 905
4-613. CCCBCNTVAL Register Field Descriptions 906
4-614. CCCABERRSTAT Register Field Descriptions 907
4-615. USERMODEEN Register Field Descriptions 908
4-616. EPWMCFG Register Field Descriptions 909
4-617. DMMSWINT0 Register Field Descriptions 910
4-618. DMMSWINT1 Register Field Descriptions 911
4-619. DMMSWINTSEL0 Register Field Descriptions 912

4-620. DMMSWINTSEL1 Register Field Descriptions 913
4-621. CCCBWDEN Register Field Descriptions 914
4-622. GPIOINTREDGESEL Register Field Descriptions 915
4-623. PWMDMATRIGEN Register Field Descriptions 916
4-624. JTAGTXDATA Register Field Descriptions 917
4-625. JTAGTXCONTROL Register Field Descriptions 918
4-626. JTAGRXDATA Register Field Descriptions 919
4-627. JTAGTXRXACK Register Field Descriptions 920
4-628. JTAGRXCONTROL Register Field Descriptions 921
4-629. MSS2GEMSWIRQ Register Field Descriptions 922
4-630. CSETBFLUSH Register Field Descriptions 923
4-631. DSS_REG Registers 924
4-632. DSS_REG Access Type Codes 926
4-633. RTIEVENTCAPTURESEL Register Field Descriptions 927
4-634. CQCFG1 Register Field Descriptions 928
4-635. TPCCPARSTATCFG Register Field Descriptions 929
4-636. TPTC0WRMPUSTADD0 Register Field Descriptions 930
4-637. TPTC0WRMPUSTADD1 Register Field Descriptions 931
4-638. TPTC0WRMPUSTADD2 Register Field Descriptions 932
4-639. TPTC0WRMPUSTADD3 Register Field Descriptions 933
4-640. TPTC0WRMPUSTADD4 Register Field Descriptions 934
4-641. TPTC0WRMPUSTADD5 Register Field Descriptions 935
4-642. TPTC0WRMPUENDADD0 Register Field Descriptions 936
4-643. TPTC0WRMPUENDADD1 Register Field Descriptions 937
4-644. TPTC0WRMPUENDADD2 Register Field Descriptions 938
4-645. TPTC0WRMPUENDADD3 Register Field Descriptions 939
4-646. TPTC0WRMPUENDADD4 Register Field Descriptions 940
4-647. TPTC0WRMPUENDADD5 Register Field Descriptions 941
4-648. TPTC0WRMPUERRADD Register Field Descriptions 942
4-649. TPTC0RDMPUSTADD0 Register Field Descriptions 943
4-650. TPTC0RDMPUSTADD1 Register Field Descriptions 944
4-651. TPTC0RDMPUSTADD2 Register Field Descriptions 945
4-652. TPTC0RDMPUSTADD3 Register Field Descriptions 946
4-653. TPTC0RDMPUSTADD4 Register Field Descriptions 947
4-654. TPTC0RDMPUSTADD5 Register Field Descriptions 948
4-655. TPTC0RDMPUENDADD0 Register Field Descriptions 949
4-656. TPTC0RDMPUENDADD1 Register Field Descriptions 950
4-657. TPTC0RDMPUENDADD2 Register Field Descriptions 951
4-658. TPTC0RDMPUENDADD3 Register Field Descriptions 952
4-659. TPTC0RDMPUENDADD4 Register Field Descriptions 953
4-660. TPTC0RDMPUENDADD5 Register Field Descriptions 954
4-661. TPTC0RDMPUERRADD Register Field Descriptions 955
4-662. TPTC1WRMPUSTADD0 Register Field Descriptions 956
4-663. TPTC1WRMPUSTADD1 Register Field Descriptions 957
4-664. TPTC1WRMPUSTADD2 Register Field Descriptions 958
4-665. TPTC1WRMPUSTADD3 Register Field Descriptions 959
4-666. TPTC1WRMPUSTADD4 Register Field Descriptions 960
4-667. TPTC1WRMPUSTADD5 Register Field Descriptions 961
4-668. TPTC1WRMPUENDADD0 Register Field Descriptions 962

4-669. TPTC1WRMPUENDADD1 Register Field Descriptions 963
4-670. TPTC1WRMPUENDADD2 Register Field Descriptions 964
4-671. TPTC1WRMPUENDADD3 Register Field Descriptions 965
4-672. TPTC1WRMPUENDADD4 Register Field Descriptions 966
4-673. TPTC1WRMPUENDADD5 Register Field Descriptions 967
4-674. TPTC1WRMPUERRADD Register Field Descriptions 968
4-675. TPTC1RDMPUSTADD0 Register Field Descriptions 969
4-676. TPTC1RDMPUSTADD1 Register Field Descriptions 970
4-677. TPTC1RDMPUSTADD2 Register Field Descriptions 971
4-678. TPTC1RDMPUSTADD3 Register Field Descriptions 972
4-679. TPTC1RDMPUSTADD4 Register Field Descriptions 973
4-680. TPTC1RDMPUSTADD5 Register Field Descriptions 974
4-681. TPTC1RDMPUENDADD0 Register Field Descriptions 975
4-682. TPTC1RDMPUENDADD1 Register Field Descriptions 976
4-683. TPTC1RDMPUENDADD2 Register Field Descriptions 977
4-684. TPTC1RDMPUENDADD3 Register Field Descriptions 978
4-685. TPTC1RDMPUENDADD4 Register Field Descriptions 979
4-686. TPTC1RDMPUENDADD5 Register Field Descriptions 980
4-687. TPTC1RDMPUERRADD Register Field Descriptions 981
4-688. TPTCMPUVALIDCFG Register Field Descriptions 982
4-689. TPTCMPUENCFG Register Field Descriptions 983
4-690. TESTPATTERNRX1ICFG Register Field Descriptions 984
4-691. TESTPATTERNRX2ICFG Register Field Descriptions 985
4-692. TESTPATTERNRX3ICFG Register Field Descriptions 986
4-693. TESTPATTERNRX4ICFG Register Field Descriptions 987
4-694. TESTPATTERNRX1QCFG Register Field Descriptions 988
4-695. TESTPATTERNRX2QCFG Register Field Descriptions 989
4-696. TESTPATTERNRX3QCFG Register Field Descriptions 990
4-697. TESTPATTERNRX4QCFG Register Field Descriptions 991
4-698. TESTPATTERNVLDCFG Register Field Descriptions 992
4-699. TPCC1PARSTATCFG Register Field Descriptions 993
4-700. DMMSWINT1 Register Field Descriptions 994
4-701. DSSINTRCFG Register Field Descriptions 995
4-702. MPUMSTIDCFG1 Register Field Descriptions 996
4-703. MPUMSTIDCFG2 Register Field Descriptions 997
4-704. MPUMSTIDCFG3 Register Field Descriptions 998
4-705. HSRAM1ECCCFG Register Field Descriptions 999
4-706. DATATRRAMECCCFG Register Field Descriptions 1000
4-707. ADCBUFPINGECCCFG Register Field Descriptions 1001
4-708. ADCBUFPONGECCCFG Register Field Descriptions 1002
4-709. UMAP0PARITYCFG1 Register Field Descriptions 1003
4-710. UMAP0PARITYCFG2 Register Field Descriptions 1004
4-711. UMAP0PARITYCFG3 Register Field Descriptions 1005
4-712. UMAP1PARITYCFG1 Register Field Descriptions 1006
4-713. UMAP1PARITYCFG2 Register Field Descriptions 1007
4-714. UMAP1PARITYCFG3 Register Field Descriptions 1008
4-715. ESMGRP2MASKCFG Register Field Descriptions 1009
4-716. L2MEMINITCFG1 Register Field Descriptions 1010
4-717. L2MEMINITCFG2 Register Field Descriptions 1012

4-718. GEMRSTCAUSE Register Field Descriptions 1013
4-719. GEMPWRSMCFG4 Register Field Descriptions 1014
4-720. PWRSMWAKEMASK0 Register Field Descriptions 1015
4-721. PWRSMWAKEMASK1 Register Field Descriptions 1016
4-722. PWRSMWAKEMASK2 Register Field Descriptions 1017
4-723. PWRSMMISEVTMASK0 Register Field Descriptions 1018
4-724. PWRSMMISEVTMASK1 Register Field Descriptions 1019
4-725. PWRSMMISEVTMASK2 Register Field Descriptions 1020
4-726. PWRSMWAKESRCSTAT0 Register Field Descriptions 1021
4-727. PWRSMWAKESRCSTAT1 Register Field Descriptions 1022
4-728. PWRSMWAKESRCSTAT2 Register Field Descriptions 1023
4-729. PWRSMEVNTMONSTAT0 Register Field Descriptions 1024
4-730. PWRSMEVNTMONSTAT1 Register Field Descriptions 1025
4-731. PWRSMEVNTMONSTAT2 Register Field Descriptions 1026
4-732. PWRSMWAKESRCSTATCLR0 Register Field Descriptions 1027
4-733. PWRSMWAKESRCSTATCLR1 Register Field Descriptions 1028
4-734. PWRSMWAKESRCSTATCLR2 Register Field Descriptions 1029
4-735. ADCBUFCFG1 Register Field Descriptions 1030
4-736. ADCBUFCFG2 Register Field Descriptions 1031
4-737. ADCBUFCFG3 Register Field Descriptions 1032
4-738. ADCBUFCFG4 Register Field Descriptions 1033
4-739. STCPBISTSMCFG1 Register Field Descriptions 1034
4-740. STCPBISTSMCFG2 Register Field Descriptions 1035
4-741. RTI2EVENTCAPTURESEL Register Field Descriptions 1036
4-742. DSSMISC5 Register Field Descriptions 1037
4-743. DSS_REG2 Registers 1038
4-744. DSS_REG2 Access Type Codes 1039
4-745. TPTC2WRMPUSTADD0 Register Field Descriptions 1040
4-746. TPTC2WRMPUSTADD1 Register Field Descriptions 1041
4-747. TPTC2WRMPUSTADD2 Register Field Descriptions 1042
4-748. TPTC2WRMPUSTADD3 Register Field Descriptions 1043
4-749. TPTC2WRMPUSTADD4 Register Field Descriptions 1044
4-750. TPTC2WRMPUSTADD5 Register Field Descriptions 1045
4-751. TPTC2WRMPUENDADD0 Register Field Descriptions 1046
4-752. TPTC2WRMPUENDADD1 Register Field Descriptions 1047
4-753. TPTC2WRMPUENDADD2 Register Field Descriptions 1048
4-754. TPTC2WRMPUENDADD3 Register Field Descriptions 1049
4-755. TPTC2WRMPUENDADD4 Register Field Descriptions 1050
4-756. TPTC2WRMPUENDADD5 Register Field Descriptions 1051
4-757. TPTC2WRMPUERRADD Register Field Descriptions 1052
4-758. TPTC2RDMPUSTADD0 Register Field Descriptions 1053
4-759. TPTC2RDMPUSTADD1 Register Field Descriptions 1054
4-760. TPTC2RDMPUSTADD2 Register Field Descriptions 1055
4-761. TPTC2RDMPUSTADD3 Register Field Descriptions 1056
4-762. TPTC2RDMPUSTADD4 Register Field Descriptions 1057
4-763. TPTC2RDMPUSTADD5 Register Field Descriptions 1058
4-764. TPTC2RDMPUENDADD0 Register Field Descriptions 1059
4-765. TPTC2RDMPUENDADD1 Register Field Descriptions 1060
4-766. TPTC2RDMPUENDADD2 Register Field Descriptions 1061

4-767. TPTC2RDMPUENDADD3 Register Field Descriptions 1062
4-768. TPTC2RDMPUENDADD4 Register Field Descriptions 1063
4-769. TPTC2RDMPUENDADD5 Register Field Descriptions 1064
4-770. TPTC2RDMPUERRADD Register Field Descriptions 1065
4-771. TPTC3WRMPUSTADD0 Register Field Descriptions 1066
4-772. TPTC3WRMPUSTADD1 Register Field Descriptions 1067
4-773. TPTC3WRMPUSTADD2 Register Field Descriptions 1068
4-774. TPTC3WRMPUSTADD3 Register Field Descriptions 1069
4-775. TPTC3WRMPUSTADD4 Register Field Descriptions 1070
4-776. TPTC3WRMPUSTADD5 Register Field Descriptions 1071
4-777. TPTC3WRMPUENDADD0 Register Field Descriptions 1072
4-778. TPTC3WRMPUENDADD1 Register Field Descriptions 1073
4-779. TPTC3WRMPUENDADD2 Register Field Descriptions 1074
4-780. TPTC3WRMPUENDADD3 Register Field Descriptions 1075
4-781. TPTC3WRMPUENDADD4 Register Field Descriptions 1076
4-782. TPTC3WRMPUENDADD5 Register Field Descriptions 1077
4-783. TPTC3WRMPUERRADD Register Field Descriptions 1078
4-784. TPTC3RDMPUSTADD0 Register Field Descriptions 1079
4-785. TPTC3RDMPUSTADD1 Register Field Descriptions 1080
4-786. TPTC3RDMPUSTADD2 Register Field Descriptions 1081
4-787. TPTC3RDMPUSTADD3 Register Field Descriptions 1082
4-788. TPTC3RDMPUSTADD4 Register Field Descriptions 1083
4-789. TPTC3RDMPUSTADD5 Register Field Descriptions 1084
4-790. TPTC3RDMPUENDADD0 Register Field Descriptions 1085
4-791. TPTC3RDMPUENDADD1 Register Field Descriptions 1086
4-792. TPTC3RDMPUENDADD2 Register Field Descriptions 1087
4-793. TPTC3RDMPUENDADD3 Register Field Descriptions 1088
4-794. TPTC3RDMPUENDADD4 Register Field Descriptions 1089
4-795. TPTC3RDMPUENDADD5 Register Field Descriptions 1090
4-796. TPTC3RDMPUERRADD Register Field Descriptions 1091
4-797. TPTCMPUVALIDCFG2 Register Field Descriptions 1092
4-798. TPTCMPUENCFG2 Register Field Descriptions 1093
4-799. L3ECCCFG1 Register Field Descriptions 1094
4-800. L3ECCCFG2 Register Field Descriptions 1095
4-801. DSS2MSSSWIRQ Register Field Descriptions 1096
4-802. MSS_TOPRCM Registers 1097
4-803. MSS_TOPRCM Access Type Codes 1098
4-804. BSSCTL Register Field Descriptions 1099
4-805. DSSCTL Register Field Descriptions 1100
4-806. EXTCLKDIV Register Field Descriptions 1101
4-807. EXTCLKSRCSEL Register Field Descriptions 1102
4-808. EXTCLKCTL Register Field Descriptions 1103
4-809. SOFTSYSRST Register Field Descriptions 1104
4-810. WDRSTEN Register Field Descriptions 1105
4-811. SYSRSTCAUSE Register Field Descriptions 1106
4-812. SYSRSTCAUSECLR Register Field Descriptions 1107
4-813. MISCCAPT Register Field Descriptions 1108
4-814. DCDCCTL0 Register Field Descriptions 1109
4-815. DCDCCTL1 Register Field Descriptions 1110

4-816. USERMODEEN Register Field Descriptions 1111
4-817. LVDSPADCTL0 Register Field Descriptions 1112
4-818. LVDSPADCTL1 Register Field Descriptions 1113
4-819. DFTREG0 Register Field Descriptions 1114
4-820. DFTREG1 Register Field Descriptions 1115
4-821. DFTREG5 Register Field Descriptions 1116
4-822. MEMINITDONE Register Field Descriptions 1117
4-823. SPARE0_3 Register Field Descriptions 1118
4-824. MSS_SIGNATURE Register Field Descriptions 1119
4-825. GEMBOOTSTCEN Register Field Descriptions 1120
4-826. MISCCTL1 Register Field Descriptions 1121
4-827. USERMODEEN2 Register Field Descriptions 1122
4-828. SYSTICK Register Field Descriptions 1123
4-829. SECURECFGREG1 Register Field Descriptions 1124
4-830. SECURECFGREG2 Register Field Descriptions 1125
4-831. SECURECFGREG3 Register Field Descriptions 1126
4-832. SECURECFGREG4 Register Field Descriptions 1127
4-833. SECURERAMREG Register Field Descriptions 1128
4-834. SPAREMULTIBIT Register Field Descriptions 1129
4-835. SPARE4_9 Register Field Descriptions 1130
4-836. UID31TO0 Register Field Descriptions 1131
4-837. UID63TO32 Register Field Descriptions 1132
4-838. UID95TO64 Register Field Descriptions 1133
4-839. UID119TO96 Register Field Descriptions 1134
4-840. MEMINITSTARTSHMEM Register Field Descriptions 1135
4-841. MEMINITDONESHMEM Register Field Descriptions 1136
4-842. DSSMEMTAB0 Register Field Descriptions 1137
4-843. TCMAMEMTAB Register Field Descriptions 1138
4-844. TCMBMEMTAB Register Field Descriptions 1139
4-845. SHMEMBANKSEL3TO0 Register Field Descriptions 1140
4-846. SHMEMBANKSEL7TO4 Register Field Descriptions 1141
4-847. PBISTCLKCTL Register Field Descriptions 1142
4-848. MSS_RCM Registers 1143
4-849. MSS_RCM Access Type Codes 1144
4-850. SOFTRST1 Register Field Descriptions 1145
4-851. SOFTRST2 Register Field Descriptions 1146
4-852. CLKDIVCTL0 Register Field Descriptions 1147
4-853. CLKSRCSEL0 Register Field Descriptions 1148
4-854. CR4CTL Register Field Descriptions 1149
4-855. CLKGATE Register Field Descriptions 1150
4-856. CLKSRCSEL1 Register Field Descriptions 1151
4-857. CURRCLKDIV0 Register Field Descriptions 1152
4-858. RTICURRCLKDIV Register Field Descriptions 1153
4-859. MEMINITSTART Register Field Descriptions 1154
4-860. CURRCLKDIV1 Register Field Descriptions 1155
4-861. MEMINITDONE Register Field Descriptions 1156
4-862. ECCENMSSGEM Register Field Descriptions 1157
4-863. ECCCAPTMSSGEM Register Field Descriptions 1158
4-864. ECCENBSSGEM Register Field Descriptions 1159

4-865. ECCCAPTBSSGEM Register Field Descriptions 1160
4-866. USERMODEEN Register Field Descriptions 1161
4-867. NSYSPERUSERMODEN Register Field Descriptions 1162
4-868. SECURERAMMMI Register Field Descriptions 1163
4-869. SECURERAMECC Register Field Descriptions 1164
4-870. ESMGATE0 Register Field Descriptions 1165
4-871. ESMGATE1 Register Field Descriptions 1166
4-872. ESMGATE2 Register Field Descriptions 1167
4-873. ESMGATE3 Register Field Descriptions 1168
4-874. ESMGATE4 Register Field Descriptions 1169
4-875. KEY Register Field Descriptions 1170
4-876. SWIRQA Register Field Descriptions 1171
4-877. SWIRQB Register Field Descriptions 1172
4-878. MISCCTL0 Register Field Descriptions 1173
4-879. ATCMERRCAPTCTL Register Field Descriptions 1174
4-880. B0TCMERRCAPTCTL Register Field Descriptions 1175
4-881. B1TCMERRCAPTCTL Register Field Descriptions 1176
4-882. SOFTCORERST Register Field Descriptions 1177
4-883. RSTCAUSE Register Field Descriptions 1178
4-884. RSTCAUSECLR Register Field Descriptions 1179
4-885. SPITRIGSRC Register Field Descriptions 1180
4-886. CLKINUSE Register Field Descriptions 1181
4-887. ECCENMSSBSS Register Field Descriptions 1182
4-888. ECCCAPTMSSBSS Register Field Descriptions 1183
4-889. CLKDIVCTL2 Register Field Descriptions 1184
4-890. SWIRQC Register Field Descriptions 1185
4-891. MSS_GPCFG_REG Registers 1186
4-892. MSS_GPCFG_REG Access Type Codes 1186
4-893. GPCFG0 Register Field Descriptions 1188
4-894. GPCFG1 Register Field Descriptions 1189
4-895. GPCFG2 Register Field Descriptions 1190
4-896. GPCFG3 Register Field Descriptions 1191
4-897. GPCFG4 Register Field Descriptions 1192
4-898. GPCFG6 Register Field Descriptions 1193
4-899. GPCFG11 Register Field Descriptions 1194
4-900. CCCACFG0 Register Field Descriptions 1195
4-901. CCCACFG1 Register Field Descriptions 1196
4-902. CCCACFG2 Register Field Descriptions 1197
4-903. CCCACFG3 Register Field Descriptions 1198
4-904. CCCBCFG0 Register Field Descriptions 1199
4-905. CCCBCFG1 Register Field Descriptions 1200
4-906. CCCBCFG2 Register Field Descriptions 1201
4-907. CCCBCFG3 Register Field Descriptions 1202
4-908. CCCACNTVAL Register Field Descriptions 1203
4-909. CCCBCNTVAL Register Field Descriptions 1204
4-910. CCCABERRSTAT Register Field Descriptions 1205
4-911. USERMODEEN Register Field Descriptions 1206
4-912. EPWMCFG Register Field Descriptions 1207
4-913. DMMSWINT0 Register Field Descriptions 1208

4-914. DMMSWINT1 Register Field Descriptions 1209
4-915. DMMSWINTSEL0 Register Field Descriptions 1210
4-916. DMMSWINTSEL1 Register Field Descriptions 1211
4-917. CCCBWDEN Register Field Descriptions 1212
4-918. GPIOINTREDGESEL Register Field Descriptions 1213
4-919. PWMDMATRIGEN Register Field Descriptions 1214
4-920. JTAGTXDATA Register Field Descriptions 1215
4-921. JTAGTXCONTROL Register Field Descriptions 1216
4-922. JTAGRXDATA Register Field Descriptions 1217
4-923. JTAGTXRXACK Register Field Descriptions 1218
4-924. JTAGRXCONTROL Register Field Descriptions 1219
4-925. MSS2GEMSWIRQ Register Field Descriptions 1220
4-926. CSETBFLUSH Register Field Descriptions 1221
4-927. DSS_REG Registers 1222
4-928. DSS_REG Access Type Codes 1224
4-929. RTIEVENTCAPTURESEL Register Field Descriptions 1225
4-930. CQCFG1 Register Field Descriptions 1226
4-931. TPCCPARSTATCFG Register Field Descriptions 1227
4-932. TPTC0WRMPUSTADD0 Register Field Descriptions 1228
4-933. TPTC0WRMPUSTADD1 Register Field Descriptions 1229
4-934. TPTC0WRMPUSTADD2 Register Field Descriptions 1230
4-935. TPTC0WRMPUSTADD3 Register Field Descriptions 1231
4-936. TPTC0WRMPUSTADD4 Register Field Descriptions 1232
4-937. TPTC0WRMPUSTADD5 Register Field Descriptions 1233
4-938. TPTC0WRMPUENDADD0 Register Field Descriptions 1234
4-939. TPTC0WRMPUENDADD1 Register Field Descriptions 1235
4-940. TPTC0WRMPUENDADD2 Register Field Descriptions 1236
4-941. TPTC0WRMPUENDADD3 Register Field Descriptions 1237
4-942. TPTC0WRMPUENDADD4 Register Field Descriptions 1238
4-943. TPTC0WRMPUENDADD5 Register Field Descriptions 1239
4-944. TPTC0WRMPUERRADD Register Field Descriptions 1240
4-945. TPTC0RDMPUSTADD0 Register Field Descriptions 1241
4-946. TPTC0RDMPUSTADD1 Register Field Descriptions 1242
4-947. TPTC0RDMPUSTADD2 Register Field Descriptions 1243
4-948. TPTC0RDMPUSTADD3 Register Field Descriptions 1244
4-949. TPTC0RDMPUSTADD4 Register Field Descriptions 1245
4-950. TPTC0RDMPUSTADD5 Register Field Descriptions 1246
4-951. TPTC0RDMPUENDADD0 Register Field Descriptions 1247
4-952. TPTC0RDMPUENDADD1 Register Field Descriptions 1248
4-953. TPTC0RDMPUENDADD2 Register Field Descriptions 1249
4-954. TPTC0RDMPUENDADD3 Register Field Descriptions 1250
4-955. TPTC0RDMPUENDADD4 Register Field Descriptions 1251
4-956. TPTC0RDMPUENDADD5 Register Field Descriptions 1252
4-957. TPTC0RDMPUERRADD Register Field Descriptions 1253
4-958. TPTC1WRMPUSTADD0 Register Field Descriptions 1254
4-959. TPTC1WRMPUSTADD1 Register Field Descriptions 1255
4-960. TPTC1WRMPUSTADD2 Register Field Descriptions 1256
4-961. TPTC1WRMPUSTADD3 Register Field Descriptions 1257
4-962. TPTC1WRMPUSTADD4 Register Field Descriptions 1258

4-963. TPTC1WRMPUSTADD5 Register Field Descriptions 1259
4-964. TPTC1WRMPUENDADD0 Register Field Descriptions 1260
4-965. TPTC1WRMPUENDADD1 Register Field Descriptions 1261
4-966. TPTC1WRMPUENDADD2 Register Field Descriptions 1262
4-967. TPTC1WRMPUENDADD3 Register Field Descriptions 1263
4-968. TPTC1WRMPUENDADD4 Register Field Descriptions 1264
4-969. TPTC1WRMPUENDADD5 Register Field Descriptions 1265
4-970. TPTC1WRMPUERRADD Register Field Descriptions 1266
4-971. TPTC1RDMPUSTADD0 Register Field Descriptions 1267
4-972. TPTC1RDMPUSTADD1 Register Field Descriptions 1268
4-973. TPTC1RDMPUSTADD2 Register Field Descriptions 1269
4-974. TPTC1RDMPUSTADD3 Register Field Descriptions 1270
4-975. TPTC1RDMPUSTADD4 Register Field Descriptions 1271
4-976. TPTC1RDMPUSTADD5 Register Field Descriptions 1272
4-977. TPTC1RDMPUENDADD0 Register Field Descriptions 1273
4-978. TPTC1RDMPUENDADD1 Register Field Descriptions 1274
4-979. TPTC1RDMPUENDADD2 Register Field Descriptions 1275
4-980. TPTC1RDMPUENDADD3 Register Field Descriptions 1276
4-981. TPTC1RDMPUENDADD4 Register Field Descriptions 1277
4-982. TPTC1RDMPUENDADD5 Register Field Descriptions 1278
4-983. TPTC1RDMPUERRADD Register Field Descriptions 1279
4-984. TPTCMPUVALIDCFG Register Field Descriptions 1280
4-985. TPTCMPUENCFG Register Field Descriptions 1281
4-986. TESTPATTERNRX1ICFG Register Field Descriptions 1282
4-987. TESTPATTERNRX2ICFG Register Field Descriptions 1283
4-988. TESTPATTERNRX3ICFG Register Field Descriptions 1284
4-989. TESTPATTERNRX4ICFG Register Field Descriptions 1285
4-990. TESTPATTERNRX1QCFG Register Field Descriptions 1286
4-991. TESTPATTERNRX2QCFG Register Field Descriptions 1287
4-992. TESTPATTERNRX3QCFG Register Field Descriptions 1288
4-993. TESTPATTERNRX4QCFG Register Field Descriptions 1289
4-994. TESTPATTERNVLDCFG Register Field Descriptions 1290
4-995. TPCC1PARSTATCFG Register Field Descriptions 1291
4-996. DMMSWINT1 Register Field Descriptions 1292
4-997. DSSINTRCFG Register Field Descriptions 1293
4-998. MPUMSTIDCFG1 Register Field Descriptions 1294
4-999. MPUMSTIDCFG2 Register Field Descriptions 1295
4-1000. MPUMSTIDCFG3 Register Field Descriptions 1296
4-1001. HSRAM1ECCCFG Register Field Descriptions 1297
4-1002. DATATRRAMECCCFG Register Field Descriptions 1298
4-1003. ADCBUFPINGECCCFG Register Field Descriptions 1299
4-1004. ADCBUFPONGECCCFG Register Field Descriptions 1300
4-1005. UMAP0PARITYCFG1 Register Field Descriptions 1301
4-1006. UMAP0PARITYCFG2 Register Field Descriptions 1302
4-1007. UMAP0PARITYCFG3 Register Field Descriptions 1303
4-1008. UMAP1PARITYCFG1 Register Field Descriptions 1304
4-1009. UMAP1PARITYCFG2 Register Field Descriptions 1305
4-1010. UMAP1PARITYCFG3 Register Field Descriptions 1306
4-1011. ESMGRP2MASKCFG Register Field Descriptions 1307

4-1012. L2MEMINITCFG1 Register Field Descriptions 1308
4-1013. L2MEMINITCFG2 Register Field Descriptions 1310
4-1014. GEMRSTCAUSE Register Field Descriptions 1311
4-1015. GEMPWRSMCFG4 Register Field Descriptions 1312
4-1016. PWRSMWAKEMASK0 Register Field Descriptions 1313
4-1017. PWRSMWAKEMASK1 Register Field Descriptions 1314
4-1018. PWRSMWAKEMASK2 Register Field Descriptions 1315
4-1019. PWRSMMISEVTMASK0 Register Field Descriptions 1316
4-1020. PWRSMMISEVTMASK1 Register Field Descriptions 1317
4-1021. PWRSMMISEVTMASK2 Register Field Descriptions 1318
4-1022. PWRSMWAKESRCSTAT0 Register Field Descriptions 1319
4-1023. PWRSMWAKESRCSTAT1 Register Field Descriptions 1320
4-1024. PWRSMWAKESRCSTAT2 Register Field Descriptions 1321
4-1025. PWRSMEVNTMONSTAT0 Register Field Descriptions 1322
4-1026. PWRSMEVNTMONSTAT1 Register Field Descriptions 1323
4-1027. PWRSMEVNTMONSTAT2 Register Field Descriptions 1324
4-1028. PWRSMWAKESRCSTATCLR0 Register Field Descriptions 1325
4-1029. PWRSMWAKESRCSTATCLR1 Register Field Descriptions 1326
4-1030. PWRSMWAKESRCSTATCLR2 Register Field Descriptions 1327
4-1031. ADCBUFCFG1 Register Field Descriptions 1328
4-1032. ADCBUFCFG2 Register Field Descriptions 1329
4-1033. ADCBUFCFG3 Register Field Descriptions 1330
4-1034. ADCBUFCFG4 Register Field Descriptions 1331
4-1035. STCPBISTSMCFG1 Register Field Descriptions 1332
4-1036. STCPBISTSMCFG2 Register Field Descriptions 1333
4-1037. RTI2EVENTCAPTURESEL Register Field Descriptions 1334
4-1038. DSSMISC5 Register Field Descriptions 1335
4-1039. DSS_REG2 Registers 1336
4-1040. DSS_REG2 Access Type Codes 1337
4-1041. TPTC2WRMPUSTADD0 Register Field Descriptions 1338
4-1042. TPTC2WRMPUSTADD1 Register Field Descriptions 1339
4-1043. TPTC2WRMPUSTADD2 Register Field Descriptions 1340
4-1044. TPTC2WRMPUSTADD3 Register Field Descriptions 1341
4-1045. TPTC2WRMPUSTADD4 Register Field Descriptions 1342
4-1046. TPTC2WRMPUSTADD5 Register Field Descriptions 1343
4-1047. TPTC2WRMPUENDADD0 Register Field Descriptions 1344
4-1048. TPTC2WRMPUENDADD1 Register Field Descriptions 1345
4-1049. TPTC2WRMPUENDADD2 Register Field Descriptions 1346
4-1050. TPTC2WRMPUENDADD3 Register Field Descriptions 1347
4-1051. TPTC2WRMPUENDADD4 Register Field Descriptions 1348
4-1052. TPTC2WRMPUENDADD5 Register Field Descriptions 1349
4-1053. TPTC2WRMPUERRADD Register Field Descriptions 1350
4-1054. TPTC2RDMPUSTADD0 Register Field Descriptions 1351
4-1055. TPTC2RDMPUSTADD1 Register Field Descriptions 1352
4-1056. TPTC2RDMPUSTADD2 Register Field Descriptions 1353
4-1057. TPTC2RDMPUSTADD3 Register Field Descriptions 1354
4-1058. TPTC2RDMPUSTADD4 Register Field Descriptions 1355
4-1059. TPTC2RDMPUSTADD5 Register Field Descriptions 1356
4-1060. TPTC2RDMPUENDADD0 Register Field Descriptions 1357

4-1061. TPTC2RDMPUENDADD1 Register Field Descriptions 1358
4-1062. TPTC2RDMPUENDADD2 Register Field Descriptions 1359
4-1063. TPTC2RDMPUENDADD3 Register Field Descriptions 1360
4-1064. TPTC2RDMPUENDADD4 Register Field Descriptions 1361
4-1065. TPTC2RDMPUENDADD5 Register Field Descriptions 1362
4-1066. TPTC2RDMPUERRADD Register Field Descriptions 1363
4-1067. TPTC3WRMPUSTADD0 Register Field Descriptions 1364
4-1068. TPTC3WRMPUSTADD1 Register Field Descriptions 1365
4-1069. TPTC3WRMPUSTADD2 Register Field Descriptions 1366
4-1070. TPTC3WRMPUSTADD3 Register Field Descriptions 1367
4-1071. TPTC3WRMPUSTADD4 Register Field Descriptions 1368
4-1072. TPTC3WRMPUSTADD5 Register Field Descriptions 1369
4-1073. TPTC3WRMPUENDADD0 Register Field Descriptions 1370
4-1074. TPTC3WRMPUENDADD1 Register Field Descriptions 1371
4-1075. TPTC3WRMPUENDADD2 Register Field Descriptions 1372
4-1076. TPTC3WRMPUENDADD3 Register Field Descriptions 1373
4-1077. TPTC3WRMPUENDADD4 Register Field Descriptions 1374
4-1078. TPTC3WRMPUENDADD5 Register Field Descriptions 1375
4-1079. TPTC3WRMPUERRADD Register Field Descriptions 1376
4-1080. TPTC3RDMPUSTADD0 Register Field Descriptions 1377
4-1081. TPTC3RDMPUSTADD1 Register Field Descriptions 1378
4-1082. TPTC3RDMPUSTADD2 Register Field Descriptions 1379
4-1083. TPTC3RDMPUSTADD3 Register Field Descriptions 1380
4-1084. TPTC3RDMPUSTADD4 Register Field Descriptions 1381
4-1085. TPTC3RDMPUSTADD5 Register Field Descriptions 1382
4-1086. TPTC3RDMPUENDADD0 Register Field Descriptions 1383
4-1087. TPTC3RDMPUENDADD1 Register Field Descriptions 1384
4-1088. TPTC3RDMPUENDADD2 Register Field Descriptions 1385
4-1089. TPTC3RDMPUENDADD3 Register Field Descriptions 1386
4-1090. TPTC3RDMPUENDADD4 Register Field Descriptions 1387
4-1091. TPTC3RDMPUENDADD5 Register Field Descriptions 1388
4-1092. TPTC3RDMPUERRADD Register Field Descriptions 1389
4-1093. TPTCMPUVALIDCFG2 Register Field Descriptions 1390
4-1094. TPTCMPUENCFG2 Register Field Descriptions 1391
4-1095. L3ECCCFG1 Register Field Descriptions 1392
4-1096. L3ECCCFG2 Register Field Descriptions 1393
4-1097. DSS2MSSSWIRQ Register Field Descriptions 1394
7-1. Arbitration According to Priority Queues and Priority Schemes 1410
7-2. Maximum Number of DMA Transactions per Channel in Non-Bypass Mode 1418
7-3. Maximum Number of DMA Transactions per Channel in Bypass Mode 1418
7-4. Control Packet RAM 1421
7-5. Control Packet RAM 1421
7-6. Parity RAM 1421
7-7. DMA Control Registers 1423
7-8. MSS_DMA_REG Access Type Codes 1424
7-9. GCTRL Register Field Descriptions 1425
7-10. PEND Register Field Descriptions 1426
7-11. FBREG Register Field Descriptions 1427
7-12. DMASTAT Register Field Descriptions 1428

7-13. HWCHENAS Register Field Descriptions 1429
7-14. HWCHENAR Register Field Descriptions 1430
7-15. SWCHENAS Register Field Descriptions 1431
7-16. SWCHENAR Register Field Descriptions 1432
7-17. CHPRIOS Register Field Descriptions 1433
7-18. CHPRIOR Register Field Descriptions 1434
7-19. GCHIENAS Register Field Descriptions 1435
7-20. GCHIENAR Register Field Descriptions 1436
7-21. DREQASI0 Register Field Descriptions 1437
7-22. DREQASI1 Register Field Descriptions 1438
7-23. DREQASI2 Register Field Descriptions 1439
7-24. DREQASI3 Register Field Descriptions 1440
7-25. DREQASI4 Register Field Descriptions 1441
7-26. DREQASI5 Register Field Descriptions 1442
7-27. DREQASI6 Register Field Descriptions 1443
7-28. DREQASI7 Register Field Descriptions 1444
7-29. PAR0 Register Field Descriptions 1445
7-30. PAR1 Register Field Descriptions 1446
7-31. PAR2 Register Field Descriptions 1447
7-32. PAR3 Register Field Descriptions 1448
7-33. FTCMAP Register Field Descriptions 1449
7-34. LFSMAP Register Field Descriptions 1450
7-35. HBCMAP Register Field Descriptions 1451
7-36. BTCMAP Register Field Descriptions 1452
7-37. BERMAP Register Field Descriptions 1453
7-38. FTCINTENAS Register Field Descriptions 1454
7-39. FTCINTENAR Register Field Descriptions 1455
7-40. LFSINTENAS Register Field Descriptions 1456
7-41. LFSINTENAR Register Field Descriptions 1457
7-42. HBCINTENAS Register Field Descriptions 1458
7-43. HBCINTENAR Register Field Descriptions 1459
7-44. BTCINTENAS Register Field Descriptions 1460
7-45. BTCINTENAR Register Field Descriptions 1461
7-46. GINTFLAG Register Field Descriptions 1462
7-47. FTCFLAG Register Field Descriptions 1463
7-48. LFSFLAG Register Field Descriptions 1464
7-49. HBCFLAG Register Field Descriptions 1465
7-50. BTCFLAG Register Field Descriptions 1466
7-51. BERFLAG Register Field Descriptions 1467
7-52. FTCAOFFSET Register Field Descriptions 1468
7-53. LFSAOFFSET Register Field Descriptions 1469
7-54. HBCAOFFSET Register Field Descriptions 1470
7-55. BTCAOFFSET Register Field Descriptions 1471
7-56. BERAOFFSET Register Field Descriptions 1472
7-57. FTCBOFFSET Register Field Descriptions 1473
7-58. LFSBOFFSET Register Field Descriptions 1474
7-59. HBCBOFFSET Register Field Descriptions 1475
7-60. BTCBOFFSET Register Field Descriptions 1476
7-61. BERBOFFSET Register Field Descriptions 1477

7-62. PTCRL Register Field Descriptions 1478
7-63. RTCTRL Register Field Descriptions 1480
7-64. DCTRL Register Field Descriptions 1481
7-65. WPR Register Field Descriptions 1482
7-66. WMR Register Field Descriptions 1483
7-67. PAACSADDR Register Field Descriptions 1484
7-68. PAACDADDR Register Field Descriptions 1485
7-69. PAACTC Register Field Descriptions 1486
7-70. PBACSADDR Register Field Descriptions 1487
7-71. PBACDADDR Register Field Descriptions 1488
7-72. PBACTC Register Field Descriptions 1489
7-73. DMAPCR Register Field Descriptions 1490
7-74. DMAPAR Register Field Descriptions 1491
7-75. DMAMPCTRL Register Field Descriptions 1492
7-76. DMAMPST Register Field Descriptions 1494
7-77. DMAMPR0S Register Field Descriptions 1495
7-78. DMAMPR0E Register Field Descriptions 1496
7-79. DMAMPR1S Register Field Descriptions 1497
7-80. DMAMPR1E Register Field Descriptions 1498
7-81. DMAMPR2S Register Field Descriptions 1499
7-82. DMAMPR2E Register Field Descriptions 1500
7-83. DMAMPR3S Register Field Descriptions 1501
7-84. DMAMPR3E Register Field Descriptions 1502
7-85. Initial Source Address (ISADDR) Field Descriptions 1503
7-86. Initial Destination Address Register (IDADDR) Field Descriptions 1503
7-87. Initial Transfer Count Register (ITCOUNT) Field Descriptions 1504
7-88. Channel Control Register (CHCTRL) Field Descriptions 1505
7-89. Element Index Offset Register (EIOFF) Field Descriptions 1506
7-90. Frame Index Offset Register (FIOFF) Field Descriptions 1506
7-91. Current Source Address Register (CSADDR) Field Descriptions 1507
7-92. Current Destination Address Register (CDADDR) Field Descriptions 1507
7-93. Current Transfer Count Register (CTCOUNT) Field Descriptions 1508
8-1. ECC Syndrome Table 1519
8-2. ECC Error Bits for Syndrome Decode 1519
8-3. CPU Reads - Address Bit 10 Selects Between Normal Data and ECC Bits 1520
8-4. CPU Writes - Address Bit 10 Selects Between Normal Data and ECC Bits 1520
8-5. VIM Control Registers 1527
8-6. Interrupt Vector Table ECC Status Register (ECCSTAT) Field Descriptions 1528
8-7. Interrupt Vector Table ECC Control Register (ECCCTL) Field Descriptions 1529
8-8. Uncorrectable Error Address Register (UERRADDR) Field Descriptions 1530
8-9. Fallback Vector Address Register (FBVECADDR) Field Descriptions 1530
8-10. Single Bit Error Address Register (SBERRADDR) Field Descriptions 1531
8-11. Interrupt Dispatch 1531
8-12. IRQ Index Offset Vector Register (IRQINDEX) Field Descriptions 1532
8-13. FIQ Index Offset Vector Register (FIQINDEX) Field Descriptions 1532
8-14. FIQ/IRQ Program Control Registers (FIRQPR) Field Descriptions 1533
8-15. Pending Interrupt Read Location Registers (INTREQ) Field Descriptions 1534
8-16. Interrupt Enable Set Registers (REQENASET) Field Descriptions 1535
8-17. Interrupt Enable Clear Registers (REQENACLR) Field Descriptions 1536

8-18. Wake-Up Enable Set Registers (WAKEENASET) Field Descriptions 1537
8-19. Wake-Up Enable Clear Registers (WAKEENACLR) Field Descriptions 1538
8-20. IRQ Interrupt Vector Register (IRQVECREG) Field Descriptions 1539
8-21. FIQ Interrupt Vector Register (FIQVECREG) Field Descriptions 1539
8-22. Capture Event Register (CAPEVT) Field Descriptions 1540
8-23. Interrupt Control Registers Organization 1541
8-24. Interrupt Control Registers (CHANCTRL[0:31]) Field Descriptions 1541
10-1. MSS_TOPRCM Registers 1550
10-2. MSS_TOPRCM Registers 1550
10-3. Register Programming 1552
10-4. Register Programming 1553
12-1. EDMA Parameter RAM Contents 1565
12-2. EDMA Channel Parameter Description 1567
12-3. Dummy and Null Transfer Request 1571
12-4. Parameter Updates in EDMA_TPCC (for Non-Null, Non-Dummy PaRAM Set) 1572
12-5. Expected Number of Transfers for Non-Null Transfer 1579
12-6. Shadow Region Registers 1583
12-7. Chain Event Triggers 1586
12-8. EDMA Transfer Completion Interrupts 1586
12-9. EDMA Error Interrupts 1586
12-10. Transfer Complete Code (TCC) to EDMA_TPCC Interrupt Mapping 1587
12-11. Number of Interrupts 1588
12-12. Allowed Accesses 1594
12-13. MPPA Registers to Region Assignment 1594
12-14. Example Access Denied 1595
12-15. Example Access Allowed 1597
12-16. Read/Write Command Optimization Rules 1601
12-17. Debug Checklist 1614
12-18. EDMA_TPCC Registers 1617
12-19. EDMA_TPCC_PID Register Field Descriptions 1620
12-20. EDMA_TPCC_CCCFG Register Field Descriptions 1621
12-21. EDMA_TPCC_QCHMAPN Register Field Descriptions 1623
12-22. EDMA_TPCC_DMAQNUMN Register Field Descriptions 1624
12-23. EDMA_TPCC_QDMAQNUM Register Field Descriptions 1625
12-24. EDMA_TPCC_QUETCMAP Register Field Descriptions 1626
12-25. EDMA_TPCC_QUEPRI Register Field Descriptions 1627
12-26. EDMA_TPCC_EMR Register Field Descriptions 1628
12-27. EDMA_TPCC_EMRH Register Field Descriptions 1630
12-28. EDMA_TPCC_EMCR Register Field Descriptions 1632
12-29. EDMA_TPCC_EMCRH Register Field Descriptions 1634
12-30. EDMA_TPCC_QEMR Register Field Descriptions 1636
12-31. EDMA_TPCC_QEMCR Register Field Descriptions 1637
12-32. EDMA_TPCC_CCERR Register Field Descriptions 1638
12-33. EDMA_TPCC_CCERRCLR Register Field Descriptions 1640
12-34. EDMA_TPCC_EEVAL Register Field Descriptions 1641
12-35. EDMA_TPCC_DRAEN Register Field Descriptions 1642
12-36. EDMA_TPCC_DRAEM Register Field Descriptions 1643
12-37. EDMA_TPCC_DRAEHM Register Field Descriptions 1645
12-38. EDMA_TPCC_QRAEN Register Field Descriptions 1647

12-39. EDMA_TPCC_QNEM Register Field Descriptions 1648
12-40. EDMA_TPCC_QNE0 Register Field Descriptions 1649
12-41. EDMA_TPCC_QNE1 Register Field Descriptions 1650
12-42. EDMA_TPCC_QNE2 Register Field Descriptions 1651
12-43. EDMA_TPCC_QNE3 Register Field Descriptions 1652
12-44. EDMA_TPCC_QNE4 Register Field Descriptions 1653
12-45. EDMA_TPCC_QNE5 Register Field Descriptions 1654
12-46. EDMA_TPCC_QNE6 Register Field Descriptions 1655
12-47. EDMA_TPCC_QNE7 Register Field Descriptions 1656
12-48. EDMA_TPCC_QNE8 Register Field Descriptions 1657
12-49. EDMA_TPCC_QNE9 Register Field Descriptions 1658
12-50. EDMA_TPCC_QNE10 Register Field Descriptions 1659
12-51. EDMA_TPCC_QNE11 Register Field Descriptions 1660
12-52. EDMA_TPCC_QNE12 Register Field Descriptions 1661
12-53. EDMA_TPCC_QNE13 Register Field Descriptions 1662
12-54. EDMA_TPCC_QNE14 Register Field Descriptions 1663
12-55. EDMA_TPCC_QNE15 Register Field Descriptions 1664
12-56. EDMA_TPCC_QSTATN Register Field Descriptions 1665
12-57. EDMA_TPCC_QWMTHRA Register Field Descriptions 1666
12-58. EDMA_TPCC_CCSTAT Register Field Descriptions 1667
12-59. EDMA_TPCC_AETCTL Register Field Descriptions 1669
12-60. EDMA_TPCC_AETSTAT Register Field Descriptions 1670
12-61. EDMA_TPCC_AETCMD Register Field Descriptions 1671
12-62. EDMA_TPCC_ER Register Field Descriptions 1672
12-63. EDMA_TPCC_ERH Register Field Descriptions 1674
12-64. EDMA_TPCC_ECR Register Field Descriptions 1676
12-65. EDMA_TPCC_ECRH Register Field Descriptions 1678
12-66. EDMA_TPCC_ESR Register Field Descriptions 1680
12-67. EDMA_TPCC_ESRH Register Field Descriptions 1682
12-68. EDMA_TPCC_CER Register Field Descriptions 1684
12-69. EDMA_TPCC_CERH Register Field Descriptions 1686
12-70. EDMA_TPCC_EER Register Field Descriptions 1688
12-71. EDMA_TPCC_EERH Register Field Descriptions 1690
12-72. EDMA_TPCC_EECR Register Field Descriptions 1692
12-73. EDMA_TPCC_EECRH Register Field Descriptions 1694
12-74. EDMA_TPCC_EESR Register Field Descriptions 1696
12-75. EDMA_TPCC_EESRH Register Field Descriptions 1698
12-76. EDMA_TPCC_SER Register Field Descriptions 1700
12-77. EDMA_TPCC_SERH Register Field Descriptions 1702
12-78. EDMA_TPCC_SECR Register Field Descriptions 1704
12-79. EDMA_TPCC_SECRH Register Field Descriptions 1706
12-80. EDMA_TPCC_IER Register Field Descriptions 1708
12-81. EDMA_TPCC_IERH Register Field Descriptions 1710
12-82. EDMA_TPCC_IECR Register Field Descriptions 1712
12-83. EDMA_TPCC_IECRH Register Field Descriptions 1714
12-84. EDMA_TPCC_IESR Register Field Descriptions 1716
12-85. EDMA_TPCC_IESRH Register Field Descriptions 1718
12-86. EDMA_TPCC_IPR Register Field Descriptions 1720
12-87. EDMA_TPCC_IPRH Register Field Descriptions 1722

12-88. EDMA_TPCC_ICR Register Field Descriptions 1724
12-89. EDMA_TPCC_ICRH Register Field Descriptions 1726
12-90. EDMA_TPCC_IEVAL Register Field Descriptions 1728
12-91. EDMA_TPCC_QER Register Field Descriptions 1729
12-92. EDMA_TPCC_QEER Register Field Descriptions 1730
12-93. EDMA_TPCC_QEECR Register Field Descriptions 1731
12-94. EDMA_TPCC_QEESR Register Field Descriptions 1732
12-95. EDMA_TPCC_QSER Register Field Descriptions 1733
12-96. EDMA_TPCC_QSECR Register Field Descriptions 1734
12-97. EDMA_TPCC_SHADOW_N Register Field Descriptions 1735
12-98. EDMA_TPCC_ER_RN Register Field Descriptions 1736
12-99. EDMA_TPCC_ERH_RN Register Field Descriptions 1738
12-100. EDMA_TPCC_ECR_RN Register Field Descriptions 1740
12-101. EDMA_TPCC_ECRH_RN Register Field Descriptions 1742
12-102. EDMA_TPCC_ESR_RN Register Field Descriptions 1744
12-103. EDMA_TPCC_ESRH_RN Register Field Descriptions 1746
12-104. EDMA_TPCC_CER_RN Register Field Descriptions 1748
12-105. EDMA_TPCC_CERH_RN Register Field Descriptions 1750
12-106. EDMA_TPCC_EER_RN Register Field Descriptions 1752
12-107. EDMA_TPCC_EERH_RN Register Field Descriptions 1754
12-108. EDMA_TPCC_EECR_RN Register Field Descriptions 1756
12-109. EDMA_TPCC_EECRH_RN Register Field Descriptions 1758
12-110. EDMA_TPCC_EESR_RN Register Field Descriptions 1760
12-111. EDMA_TPCC_EESRH_RN Register Field Descriptions 1762
12-112. EDMA_TPCC_SER_RN Register Field Descriptions 1764
12-113. EDMA_TPCC_SERH_RN Register Field Descriptions 1766
12-114. EDMA_TPCC_SECR_RN Register Field Descriptions 1768
12-115. EDMA_TPCC_SECRH_RN Register Field Descriptions 1770
12-116. EDMA_TPCC_IER_RN Register Field Descriptions 1772
12-117. EDMA_TPCC_IERH_RN Register Field Descriptions 1774
12-118. EDMA_TPCC_IECR_RN Register Field Descriptions 1776
12-119. EDMA_TPCC_IECRH_RN Register Field Descriptions 1778
12-120. EDMA_TPCC_IESR_RN Register Field Descriptions 1780
12-121. EDMA_TPCC_IESRH_RN Register Field Descriptions 1782
12-122. EDMA_TPCC_IPR_RN Register Field Descriptions 1784
12-123. EDMA_TPCC_IPRH_RN Register Field Descriptions 1786
12-124. EDMA_TPCC_ICR_RN Register Field Descriptions 1788
12-125. EDMA_TPCC_ICRH_RN Register Field Descriptions 1790
12-126. EDMA_TPCC_IEVAL_RN Register Field Descriptions 1792
12-127. EDMA_TPCC_QER_RN Register Field Descriptions 1793
12-128. EDMA_TPCC_QEER_RN Register Field Descriptions 1794
12-129. EDMA_TPCC_QEECR_RN Register Field Descriptions 1795
12-130. EDMA_TPCC_QEESR_RN Register Field Descriptions 1796
12-131. EDMA_TPCC_QSER_RN Register Field Descriptions 1797
12-132. EDMA_TPCC_QSECR_RN Register Field Descriptions 1798
12-133. EDMA_TPCC_PARAMSET Register Field Descriptions 1799
12-134. EDMA_TPCC_OPT Register Field Descriptions 1800
12-135. EDMA_TPCC_SRC Register Field Descriptions 1802
12-136. EDMA_TPCC_ABCNT Register Field Descriptions 1803

12-137. EDMA_TPCC_DST Register Field Descriptions 1804
12-138. EDMA_TPCC_BIDX Register Field Descriptions 1805
12-139. EDMA_TPCC_LNK Register Field Descriptions 1806
12-140. EDMA_TPCC_CIDX Register Field Descriptions 1807
12-141. EDMA_TPCC_CCNT Register Field Descriptions 1808
12-142. EDMA_TPTC Registers 1809
12-143. EDMA_TPTC_PID Register Field Descriptions 1810
12-144. EDMA_TPTC_TCCFG Register Field Descriptions 1811
12-145. EDMA_TPTC_TCSTAT Register Field Descriptions 1812
12-146. EDMA_TPTC_INTSTAT Register Field Descriptions 1813
12-147. EDMA_TPTC_INTEN Register Field Descriptions 1814
12-148. EDMA_TPTC_INTCLR Register Field Descriptions 1815
12-149. EDMA_TPTC_INTCMD Register Field Descriptions 1816
12-150. EDMA_TPTC_ERRSTAT Register Field Descriptions 1817
12-151. EDMA_TPTC_ERREN Register Field Descriptions 1818
12-152. EDMA_TPTC_ERRCLR Register Field Descriptions 1819
12-153. EDMA_TPTC_ERRDET Register Field Descriptions 1820
12-154. EDMA_TPTC_ERRCMD Register Field Descriptions 1821
12-155. EDMA_TPTC_RDRATE Register Field Descriptions 1822
12-156. EDMA_TPTC_POPT Register Field Descriptions 1823
12-157. EDMA_TPTC_PSRC Register Field Descriptions 1824
12-158. EDMA_TPTC_PCNT Register Field Descriptions 1825
12-159. EDMA_TPTC_PDST Register Field Descriptions 1826
12-160. EDMA_TPTC_PBIDX Register Field Descriptions 1827
12-161. EDMA_TPTC_PMPPRXY Register Field Descriptions 1828
12-162. EDMA_TPTC_SAOPT Register Field Descriptions 1829
12-163. EDMA_TPTC_SASRC Register Field Descriptions 1830
12-164. EDMA_TPTC_SACNT Register Field Descriptions 1831
12-165. EDMA_TPTC_SABIDX Register Field Descriptions 1832
12-166. EDMA_TPTC_SAMPPRXY Register Field Descriptions 1833
12-167. EDMA_TPTC_SACNTRLD Register Field Descriptions 1834
12-168. EDMA_TPTC_SASRCBREF Register Field Descriptions 1835
12-169. EDMA_TPTC_SADSTBREF Register Field Descriptions 1836
12-170. EDMA_TPTC_DFCNTRLD Register Field Descriptions 1837
12-171. EDMA_TPTC_DFSRCBREF Register Field Descriptions 1838
12-172. EDMA_TPTC_DFOPT Register Field Descriptions 1839
12-173. EDMA_TPTC_DFSRC Register Field Descriptions 1840
12-174. EDMA_TPTC_DFCNT Register Field Descriptions 1841
12-175. EDMA_TPTC_DFDST Register Field Descriptions 1842
12-176. EDMA_TPTC_DFBIDX Register Field Descriptions 1843
12-177. EDMA_TPTC_DFMPPRXY Register Field Descriptions 1844
13-1. ADC Buffer Single-Chirp and Multi-Chirp Mode Programming Sequence 1847
13-2. ADC Buffer Continuous Mode Programming Sequence 1848
13-3. Non-Interleaved Data Format 1850
14-1. Main Sequence – PRCM and Global Configuration 1859
14-2. Main Sequence – CBUFF LVDS Static Configuration 1860
14-3. Main Sequence – CBUFF Linklist 1860
14-4. DSS_CBUFF Registers 1862
14-5. DSS_CBUFF Access Type Codes 1865

14-6. CONFIG_REG_0 Register Field Descriptions 1866
14-7. CFG_SPHDR_ADDRESS Register Field Descriptions 1868
14-8. CFG_CMD_HSVAL Register Field Descriptions 1869
14-9. CFG_CMD_HEVAL Register Field Descriptions 1870
14-10. CFG_CMD_VSVAL Register Field Descriptions 1871
14-11. CFG_CMD_VEVAL Register Field Descriptions 1872
14-12. CFG_LPHDR_ADDRESS Register Field Descriptions 1873
14-13. CFG_CHIRPS_PER_FRAME Register Field Descriptions 1874
14-14. CFG_FIFO_FREE_THRESHOLD Register Field Descriptions 1875
14-15. CFG_LPPYLD_ADDRESS Register Field Descriptions 1876
14-16. CFG_DATA_LL0 Register Field Descriptions 1877
14-17. CFG_DATA_LL0_LPHDR_VAL Register Field Descriptions 1878
14-18. CFG_DATA_LL0_THRESHOLD Register Field Descriptions 1879
14-19. CFG_DATA_LL1 Register Field Descriptions 1880
14-20. CFG_DATA_LL1_LPHDR_VAL Register Field Descriptions 1881
14-21. CFG_DATA_LL1_THRESHOLD Register Field Descriptions 1882
14-22. CFG_DATA_LL2 Register Field Descriptions 1883
14-23. CFG_DATA_LL2_LPHDR_VAL Register Field Descriptions 1884
14-24. CFG_DATA_LL2_THRESHOLD Register Field Descriptions 1885
14-25. CFG_DATA_LL3 Register Field Descriptions 1886
14-26. CFG_DATA_LL3_LPHDR_VAL Register Field Descriptions 1887
14-27. CFG_DATA_LL3_THRESHOLD Register Field Descriptions 1888
14-28. CFG_DATA_LL4 Register Field Descriptions 1889
14-29. CFG_DATA_LL4_LPHDR_VAL Register Field Descriptions 1890
14-30. CFG_DATA_LL4_THRESHOLD Register Field Descriptions 1891
14-31. CFG_DATA_LL5 Register Field Descriptions 1892
14-32. CFG_DATA_LL5_LPHDR_VAL Register Field Descriptions 1893
14-33. CFG_DATA_LL5_THRESHOLD Register Field Descriptions 1894
14-34. CFG_DATA_LL6 Register Field Descriptions 1895
14-35. CFG_DATA_LL6_LPHDR_VAL Register Field Descriptions 1896
14-36. CFG_DATA_LL6_THRESHOLD Register Field Descriptions 1897
14-37. CFG_DATA_LL7 Register Field Descriptions 1898
14-38. CFG_DATA_LL7_LPHDR_VAL Register Field Descriptions 1899
14-39. CFG_DATA_LL7_THRESHOLD Register Field Descriptions 1900
14-40. CFG_DATA_LL8 Register Field Descriptions 1901
14-41. CFG_DATA_LL8_LPHDR_VAL Register Field Descriptions 1902
14-42. CFG_DATA_LL8_THRESHOLD Register Field Descriptions 1903
14-43. CFG_DATA_LL9 Register Field Descriptions 1904
14-44. CFG_DATA_LL9_LPHDR_VAL Register Field Descriptions 1905
14-45. CFG_DATA_LL9_THRESHOLD Register Field Descriptions 1906
14-46. CFG_DATA_LL10 Register Field Descriptions 1907
14-47. CFG_DATA_LL10_LPHDR_VAL Register Field Descriptions 1908
14-48. CFG_DATA_LL10_THRESHOLD Register Field Descriptions 1909
14-49. CFG_DATA_LL11 Register Field Descriptions 1910
14-50. CFG_DATA_LL11_LPHDR_VAL Register Field Descriptions 1911
14-51. CFG_DATA_LL11_THRESHOLD Register Field Descriptions 1912
14-52. CFG_DATA_LL12 Register Field Descriptions 1913
14-53. CFG_DATA_LL12_LPHDR_VAL Register Field Descriptions 1914
14-54. CFG_DATA_LL12_THRESHOLD Register Field Descriptions 1915

14-55. CFG_DATA_LL13 Register Field Descriptions 1916
14-56. CFG_DATA_LL13_LPHDR_VAL Register Field Descriptions 1917
14-57. CFG_DATA_LL13_THRESHOLD Register Field Descriptions 1918
14-58. CFG_DATA_LL14 Register Field Descriptions 1919
14-59. CFG_DATA_LL14_LPHDR_VAL Register Field Descriptions 1920
14-60. CFG_DATA_LL14_THRESHOLD Register Field Descriptions 1921
14-61. CFG_DATA_LL15 Register Field Descriptions 1922
14-62. CFG_DATA_LL15_LPHDR_VAL Register Field Descriptions 1923
14-63. CFG_DATA_LL15_THRESHOLD Register Field Descriptions 1924
14-64. CFG_DATA_LL16 Register Field Descriptions 1925
14-65. CFG_DATA_LL16_LPHDR_VAL Register Field Descriptions 1926
14-66. CFG_DATA_LL16_THRESHOLD Register Field Descriptions 1927
14-67. CFG_DATA_LL17 Register Field Descriptions 1928
14-68. CFG_DATA_LL17_LPHDR_VAL Register Field Descriptions 1929
14-69. CFG_DATA_LL17_THRESHOLD Register Field Descriptions 1930
14-70. CFG_DATA_LL18 Register Field Descriptions 1931
14-71. CFG_DATA_LL18_LPHDR_VAL Register Field Descriptions 1932
14-72. CFG_DATA_LL18_THRESHOLD Register Field Descriptions 1933
14-73. CFG_DATA_LL19 Register Field Descriptions 1934
14-74. CFG_DATA_LL19_LPHDR_VAL Register Field Descriptions 1935
14-75. CFG_DATA_LL19_THRESHOLD Register Field Descriptions 1936
14-76. CFG_DATA_LL20 Register Field Descriptions 1937
14-77. CFG_DATA_LL20_LPHDR_VAL Register Field Descriptions 1938
14-78. CFG_DATA_LL20_THRESHOLD Register Field Descriptions 1939
14-79. CFG_DATA_LL21 Register Field Descriptions 1940
14-80. CFG_DATA_LL21_LPHDR_VAL Register Field Descriptions 1941
14-81. CFG_DATA_LL21_THRESHOLD Register Field Descriptions 1942
14-82. CFG_DATA_LL22 Register Field Descriptions 1943
14-83. CFG_DATA_LL22_LPHDR_VAL Register Field Descriptions 1944
14-84. CFG_DATA_LL22_THRESHOLD Register Field Descriptions 1945
14-85. CFG_DATA_LL23 Register Field Descriptions 1946
14-86. CFG_DATA_LL23_LPHDR_VAL Register Field Descriptions 1947
14-87. CFG_DATA_LL23_THRESHOLD Register Field Descriptions 1948
14-88. CFG_DATA_LL24 Register Field Descriptions 1949
14-89. CFG_DATA_LL24_LPHDR_VAL Register Field Descriptions 1950
14-90. CFG_DATA_LL24_THRESHOLD Register Field Descriptions 1951
14-91. CFG_DATA_LL25 Register Field Descriptions 1952
14-92. CFG_DATA_LL25_LPHDR_VAL Register Field Descriptions 1953
14-93. CFG_DATA_LL25_THRESHOLD Register Field Descriptions 1954
14-94. CFG_DATA_LL26 Register Field Descriptions 1955
14-95. CFG_DATA_LL26_LPHDR_VAL Register Field Descriptions 1956
14-96. CFG_DATA_LL26_THRESHOLD Register Field Descriptions 1957
14-97. CFG_DATA_LL27 Register Field Descriptions 1958
14-98. CFG_DATA_LL27_LPHDR_VAL Register Field Descriptions 1959
14-99. CFG_DATA_LL27_THRESHOLD Register Field Descriptions 1960
14-100. CFG_DATA_LL28 Register Field Descriptions 1961
14-101. CFG_DATA_LL28_LPHDR_VAL Register Field Descriptions 1962
14-102. CFG_DATA_LL28_THRESHOLD Register Field Descriptions 1963
14-103. CFG_DATA_LL29 Register Field Descriptions 1964

14-104. CFG_DATA_LL29_LPHDR_VAL Register Field Descriptions 1965
14-105. CFG_DATA_LL29_THRESHOLD Register Field Descriptions 1966
14-106. CFG_DATA_LL30 Register Field Descriptions 1967
14-107. CFG_DATA_LL30_LPHDR_VAL Register Field Descriptions 1968
14-108. CFG_DATA_LL30_THRESHOLD Register Field Descriptions 1969
14-109. CFG_DATA_LL31 Register Field Descriptions 1970
14-110. CFG_DATA_LL31_LPHDR_VAL Register Field Descriptions 1971
14-111. CFG_DATA_LL31_THRESHOLD Register Field Descriptions 1972
14-112. CFG_LVDS_MAPPING_LANE0_FMT_0 Register Field Descriptions 1973
14-113. CFG_LVDS_MAPPING_LANE1_FMT_0 Register Field Descriptions 1974
14-114. CFG_LVDS_MAPPING_LANE2_FMT_0 Register Field Descriptions 1975
14-115. CFG_LVDS_MAPPING_LANE3_FMT_0 Register Field Descriptions 1976
14-116. CFG_LVDS_MAPPING_LANE0_FMT_1 Register Field Descriptions 1977
14-117. CFG_LVDS_MAPPING_LANE1_FMT_1 Register Field Descriptions 1978
14-118. CFG_LVDS_MAPPING_LANE2_FMT_1 Register Field Descriptions 1979
14-119. CFG_LVDS_MAPPING_LANE3_FMT_1 Register Field Descriptions 1980
14-120. CFG_LVDS_GEN_0 Register Field Descriptions 1981
14-121. CFG_LVDS_GEN_1 Register Field Descriptions 1982
14-122. CFG_LVDS_GEN_2 Register Field Descriptions 1983
14-123. CFG_MASK_REG0 Register Field Descriptions 1984
14-124. CFG_MASK_REG1 Register Field Descriptions 1985
14-125. CFG_MASK_REG2 Register Field Descriptions 1986
14-126. CFG_MASK_REG3 Register Field Descriptions 1987
14-127. STAT_CBUFF_REG0 Register Field Descriptions 1988
14-128. STAT_CBUFF_REG1 Register Field Descriptions 1989
14-129. STAT_CBUFF_REG2 Register Field Descriptions 1990
14-130. STAT_CBUFF_REG3 Register Field Descriptions 1991
14-131. CLR_CBUFF_REG0 Register Field Descriptions 1992
14-132. STAT_CBUFF_ECC_REG Register Field Descriptions 1993
14-133. MASK_CBUFF_ECC_REG Register Field Descriptions 1994
14-134. CLR_CBUFF_ECC_REG Register Field Descriptions 1995
14-135. STAT_SAFETY Register Field Descriptions 1996
14-136. MASK_SAFETY Register Field Descriptions 1997
14-137. CLR_SAFETY Register Field Descriptions 1998
14-138. I/O Description of CSI2 1999
14-139. CSI2 Lane Configuration 2000
14-140. Data Type Classes 2005
14-141. Register Access Width Limitations 2009
14-142. LP to HS Timing Parameters 2012
14-143. LP to HS Timing Parameters Example for 400-MHz DDR Clock 2012
14-144. HS to LP Timing Parameters 2014
14-145. HS to LP Timing Parameters Example for 400-MHz DDR Clock and Two Data Lanes 2014
14-146. Main Sequence – PRCM and Global Configuration 2021
14-147. Main Sequence – Configure CSI2 Clock Configuration 2022
14-148. Main Sequence – Configure CSI2_PHY 2022
14-149. Main Sequence – CSI2 Complex IO Configuration 2022
14-150. Main Sequence – VC and OCP Configuration 2023
14-151. Main Sequence – CBUFF CSI2 Static Configuration 2023
14-152. Main Sequence – CBUFF Linklist 2024

14-153. CSI2_PROTOCOL_ENGINE Registers 2025
14-154. CSI2_PROTOCOL_ENGINE Access Type Codes 2030
14-155. CSI2_REVISION Register Field Descriptions 2031
14-156. CSI2_SYSCONFIG Register Field Descriptions 2032
14-157. CSI2_SYSSTATUS Register Field Descriptions 2033
14-158. CSI2_IRQSTATUS Register Field Descriptions 2034
14-159. CSI2_IRQENABLE Register Field Descriptions 2037
14-160. CSI2_CTRL Register Field Descriptions 2039
14-161. CSI2_GNQ Register Field Descriptions 2042
14-162. CSI2_COMPLEXIO_CFG1 Register Field Descriptions 2044
14-163. CSI2_COMPLEXIO_IRQSTATUS Register Field Descriptions 2047
14-164. CSI2_COMPLEXIO_IRQENABLE Register Field Descriptions 2051
14-165. CSI2_CLK_CTRL Register Field Descriptions 2054
14-166. CSI2_TIMING1 Register Field Descriptions 2056
14-167. CSI2_TIMING2 Register Field Descriptions 2058
14-168. CSI2_VM_TIMING1 Register Field Descriptions 2060
14-169. CSI2_VM_TIMING2 Register Field Descriptions 2061
14-170. CSI2_VM_TIMING3 Register Field Descriptions 2062
14-171. CSI2_CLK_TIMING Register Field Descriptions 2063
14-172. CSI2_TX_FIFO_VC_SIZE Register Field Descriptions 2064
14-173. CSI2_RX_FIFO_VC_SIZE Register Field Descriptions 2065
14-174. CSI2_COMPLEXIO_CFG2 Register Field Descriptions 2066
14-175. CSI2_RX_FIFO_VC_FULLNESS Register Field Descriptions 2070
14-176. CSI2_VM_TIMING4 Register Field Descriptions 2071
14-177. CSI2_TX_FIFO_VC_EMPTINESS Register Field Descriptions 2072
14-178. CSI2_VM_TIMING5 Register Field Descriptions 2073
14-179. CSI2_VM_TIMING6 Register Field Descriptions 2074
14-180. CSI2_VM_TIMING7 Register Field Descriptions 2075
14-181. CSI2_STOPCLK_TIMING Register Field Descriptions 2076
14-182. CSI2_CTRL2 Register Field Descriptions 2077
14-183. CSI2_VM_TIMING8 Register Field Descriptions 2079
14-184. CSI2_TE_HSYNC_WIDTH_0 to CSI2_TE_HSYNC_WIDTH_1 Register Field Descriptions 2080
14-185. CSI2_TE_VSYNC_WIDTH_0 to CSI2_TE_VSYNC_WIDTH_1 Register Field Descriptions 2081
14-186. CSI2_TE_HSYNC_NUMBER_0 to CSI2_TE_HSYNC_NUMBER_1 Register Field Descriptions 2082
14-187. CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register Field Descriptions 2083
14-188. CSI2_VC_TE_0 to CSI2_VC_TE_3 Register Field Descriptions 2087
14-189. CSI2_VC_LONG_PACKET_HEADER_0 to CSI2_VC_LONG_PACKET_HEADER_3 Register Field Descriptions 2088
14-190. CSI2_VC_LONG_PACKET_PAYLOAD_0 to CSI2_VC_LONG_PACKET_PAYLOAD_3 Register Field Descriptions 2089
14-191. CSI2_VC_SHORT_PACKET_HEADER_0 to CSI2_VC_SHORT_PACKET_HEADER_3 Register Field Descriptions 2090
14-192. CSI2_VC_IRQSTATUS_0 to CSI2_VC_IRQSTATUS_3 Register Field Descriptions 2091
14-193. CSI2_VC_IRQENABLE_0 to CSI2_VC_IRQENABLE_3 Register Field Descriptions 2093
14-194. CSI2_PHY Registers 2095
14-195. CSI2_PHY Access Type Codes 2095
14-196. REGISTER0 Register Field Descriptions 2096
14-197. REGISTER1 Register Field Descriptions 2097
14-198. REGISTER2 Register Field Descriptions 2098
14-199. REGISTER3 Register Field Descriptions 2099

14-200. REGISTER4 Register Field Descriptions 2100
14-201. REGISTER5 Register Field Descriptions 2101
14-202. REGISTER6 Register Field Descriptions 2102
14-203. REGISTER7 Register Field Descriptions 2103
14-204. REGISTER8 Register Field Descriptions 2104
14-205. REGISTER9 Register Field Descriptions 2105
14-206. REGISTER10 Register Field Descriptions 2107
14-207. REGISTER11 Register Field Descriptions 2108
14-208. REGISTER12 Register Field Descriptions 2109
14-209. REGISTER13 Register Field Descriptions 2110
14-210. REGISTER14 Register Field Descriptions 2111
14-211. REGISTER15 Register Field Descriptions 2112
16-1. RTI Registers 2124
16-2. RTI Global Control Register (RTIGCTRL) Field Descriptions 2125
16-3. RTI Timebase Control Register (RTITBCTRL) Field Descriptions 2126
16-4. RTI Capture Control Register (RTICAPCTRL) Field Descriptions 2127
16-5. RTI Compare Control Register (RTICOMPCTRL) Field Descriptions 2128
16-6. RTI Free Running Counter 0 Register (RTIFRC0) Field Descriptions 2129
16-7. RTI Up Counter 0 Register (RTIUC0) Field Descriptions 2129
16-8. RTI Compare Up Counter 0 Register (RTICPUC0) Field Descriptions 2130
16-9. RTI Capture Free Running Counter 0 Register (RTICAFRC0) Field Descriptions 2130
16-10. RTI Capture Up Counter 0 Register (RTICAUC0) Field Descriptions 2131
16-11. RTI Free Running Counter 1 Register (RTIFRC1) Field Descriptions 2131
16-12. RTI Up Counter 1 Register (RTIUC1) Field Descriptions 2132
16-13. RTI Compare Up Counter 1 Register (RTICPUC1) Field Descriptions 2133
16-14. RTI Capture Free Running Counter 1 Register (RTICAFRC1) Field Descriptions 2134
16-15. RTI Capture Up Counter 1 Register (RTICAUC1) Field Descriptions 2134
16-16. RTI Compare 0 Register (RTICOMP0) Field Descriptions 2135
16-17. RTI Update Compare 0 Register (RTIUDCP0) Field Descriptions 2135
16-18. RTI Compare 1 Register (RTICOMP1) Field Descriptions 2136
16-19. RTI Update Compare 1 Register (RTIUDCP1) Field Descriptions 2136
16-20. RTI Compare 2 Register (RTICOMP2) Field Descriptions 2137
16-21. RTI Update Compare 2 Register (RTIUDCP2) Field Descriptions 2137
16-22. RTI Compare 3 Register (RTICOMP3) Field Descriptions 2138
16-23. RTI Update Compare 3 Register (RTIUDCP3) Field Descriptions 2138
16-24. RTI Timebase Low Compare Register (RTITBLCOMP) Field Descriptions 2139
16-25. RTI Timebase High Compare Register (RTITBHCOMP) Field Descriptions 2139
16-26. RTI Set Interrupt Control Register (RTISETINTENA) Field Descriptions 2140
16-27. RTI Clear Interrupt Control Register (RTICLEARINTENA) Field Descriptions 2142
16-28. RTI Interrupt Flag Register (RTIINTFLAG) Field Descriptions 2144
16-29. Digital Watchdog Control Register (RTIDWDCTRL) Field Descriptions 2145
16-30. Digital Watchdog Preload Register (RTIDWDPRLD) Field Descriptions 2146
16-31. Watchdog Status Register (RTIWDSTATUS) Field Descriptions 2147
16-32. RTI Watchdog Key Register (RTIDWDKEY) Field Descriptions 2148
16-33. Example of a WDKEY Sequence 2148
16-34. RTI Watchdog Down Counter Register (RTIDWDCNTR) Field Descriptions 2149
16-35. Digital Windowed Watchdog Reaction Control (RTIWWDRXNCTRL) Field Descriptions 2149
16-36. Digital Windowed Watchdog Window Size Control (RTIWWDSIZECTRL) Field Descriptions 2150
16-37. RTI Compare Interrupt Clear Enable Register (RTIINTCLRENABLE) Field Descriptions 2151

16-38. RTI Compare 0 Clear Register (RTICMP0CLR) Field Descriptions 2152
16-39. RTI Compare 1 Clear Register (RTICMP1CLR) Field Descriptions 2152
16-40. RTI Compare 2 Clear Register (RTICMP2CLR) Field Descriptions 2153
16-41. RTI Compare 3 Clear Register (RTICMP3CLR) Field Descriptions 2153
17-1. MSS_GIO Registers 2161
17-2. GIOGCR Register Field Descriptions 2163
17-3. GIOPWDN Register Field Descriptions 2164
17-4. GIOINTDET Register Field Descriptions 2165
17-5. GIOPOL Register Field Descriptions 2166
17-6. GIOENASET Register Field Descriptions 2167
17-7. GIOENACLR Register Field Descriptions 2168
17-8. GIOLVLSET Register Field Descriptions 2169
17-9. GIOLVLCLR Register Field Descriptions 2170
17-10. GIOFLG Register Field Descriptions 2171
17-11. GIOOFFA Register Field Descriptions 2172
17-12. GIOOFFB Register Field Descriptions 2173
17-13. GIOEMUA Register Field Descriptions 2174
17-14. GIOEMUB Register Field Descriptions 2175
17-15. GIODIRA Register Field Descriptions 2176
17-16. GIODINA Register Field Descriptions 2177
17-17. GIODOUTA Register Field Descriptions 2178
17-18. GIOSETA Register Field Descriptions 2179
17-19. GIOCLRA Register Field Descriptions 2180
17-20. GIOPDRA Register Field Descriptions 2181
17-21. GIOPULDISA Register Field Descriptions 2182
17-22. GIOPSLA Register Field Descriptions 2183
17-23. GIODIRB Register Field Descriptions 2184
17-24. GIODINB Register Field Descriptions 2185
17-25. GIODOUTB Register Field Descriptions 2186
17-26. GIOSETB Register Field Descriptions 2187
17-27. GIOCLRB Register Field Descriptions 2188
17-28. GIOPDRB Register Field Descriptions 2189
17-29. GIOPULDISB Register Field Descriptions 2190
17-30. GIOPSLB Register Field Descriptions 2191
17-31. GIODIRC Register Field Descriptions 2192
17-32. GIODINC Register Field Descriptions 2193
17-33. GIODOUTC Register Field Descriptions 2194
17-34. GIOSETC Register Field Descriptions 2195
17-35. GIOCLRC Register Field Descriptions 2196
17-36. GIOPDRC Register Field Descriptions 2197
17-37. GIOPULDISC Register Field Descriptions 2198
17-38. GIOPSLC Register Field Descriptions 2199
17-39. GIODIRD Register Field Descriptions 2200
17-40. GIODIND Register Field Descriptions 2201
17-41. GIODOUTD Register Field Descriptions 2202
17-42. GIOSETD Register Field Descriptions 2203
17-43. GIOCLRD Register Field Descriptions 2204
17-44. GIOPDRD Register Field Descriptions 2205
17-45. GIOPULDISD Register Field Descriptions 2206

17-46. GIOPSLD Register Field Descriptions 2207
17-47. GIODIRE Register Field Descriptions 2208
17-48. GIODINE Register Field Descriptions 2209
17-49. GIODOUTE Register Field Descriptions 2210
17-50. GIOSETE Register Field Descriptions 2211
17-51. GIOCLRE Register Field Descriptions 2212
17-52. GIOPDRE Register Field Descriptions 2213
17-53. GIOPULDISE Register Field Descriptions 2214
17-54. GIOPSLE Register Field Descriptions 2215
17-55. GIODIRF Register Field Descriptions 2216
17-56. GIODINF Register Field Descriptions 2217
17-57. GIODOUTF Register Field Descriptions 2218
17-58. GIOSETF Register Field Descriptions 2219
17-59. GIOCLRF Register Field Descriptions 2220
17-60. GIOPDRF Register Field Descriptions 2221
17-61. GIOPULDISF Register Field Descriptions 2222
17-62. GIOPSLF Register Field Descriptions 2223
17-63. GIODIRG Register Field Descriptions 2224
17-64. GIODING Register Field Descriptions 2225
17-65. GIODOUTG Register Field Descriptions 2226
17-66. GIOSETG Register Field Descriptions 2227
17-67. GIOCLRG Register Field Descriptions 2228
17-68. GIOPDRG Register Field Descriptions 2229
17-69. GIOPULDISG Register Field Descriptions 2230
17-70. GIOPSLG Register Field Descriptions 2231
17-71. GIODIRH Register Field Descriptions 2232
17-72. GIODINH Register Field Descriptions 2233
17-73. GIODOUTH Register Field Descriptions 2234
17-74. GIOSETH Register Field Descriptions 2235
17-75. GIOCLRH Register Field Descriptions 2236
17-76. GIOPDRH Register Field Descriptions 2237
17-77. GIOPULDISH Register Field Descriptions 2238
17-78. GIOPSLH Register Field Descriptions 2239
17-79. GIOSRCA Register Field Descriptions 2240
17-80. GIOSRCB Register Field Descriptions 2241
17-81. GIOSRCC Register Field Descriptions 2242
17-82. GIOSRCD Register Field Descriptions 2243
17-83. GIOSRCE Register Field Descriptions 2244
17-84. GIOSRCF Register Field Descriptions 2245
17-85. GIOSRCG Register Field Descriptions 2246
17-86. GIOSRCH Register Field Descriptions 2247
17-87. Output Buffer, and Pull Control Behavior for GIO Pins 2248
18-1. Mailbox Registers 2251
18-2. Mailbox Access Type Codes 2251
18-3. INT_MASK Register Field Descriptions 2252
18-4. INT_MASK_SET Register Field Descriptions 2253
18-5. INT_MASK_CLR Register Field Descriptions 2254
18-6. INT_STS_CLR Register Field Descriptions 2255
18-7. INT_ACK Register Field Descriptions 2256

18-8. INT_TRIG Register Field Descriptions 2257
18-9. INT_STS_MASKED Register Field Descriptions 2258
18-10. INT_STS_RAW Register Field Descriptions 2259
19-1. Encoding of Destination Bits in Trace Mode Packet Format 2263
19-2. Encoding of Status Bits in Trace Mode Packet Format 2263
19-3. Encoding of Write Size in Packet Format 2263
19-4. Number of Clock Cycles per Packet 2264
19-5. Pins Used for Data Communication 2264
19-6. DMM Registers 2267
19-7. DMM Global Control Register (DMMGLBCTRL) Field Descriptions 2268
19-8. DMM Interrupt Set Register (DMMINTSET) Field Descriptions 2270
19-9. DMM Interrupt Clear Register (DMMINTCLR) Field Descriptions 2274
19-10. DMM Interrupt Level Register (DMMINTLVL) Field Descriptions 2279
19-11. DMM Interrupt Flag Register (DMMINTFLG) Field Descriptions 2281
19-12. DMM Interrupt Offset 1 Register (DMMOFF1) Field Descriptions 2285
19-13. DMM Interrupt Offset 2 Register (DMMOFF1) Field Descriptions 2286
19-14. DMM Direct Data Mode Destination Register (DMMDDMDEST) Field Descriptions 2287
19-15. DMM Direct Data Mode Blocksize Register (DMMDDMBL) Field Descriptions 2287
19-16. DMM Direct Data Mode Pointer Register (DMMDDMPT) Field Descriptions 2288
19-17. DMM Direct Data Mode Interrupt Pointer Register (DMMINTPT) Field Descriptions 2288
19-18. DMM Destination x Region 1 (DMMDESTxREG1) Field Descriptions 2289
19-19. DMM Destination x Blocksize 1 (DMMDESTxBL1) Field Descriptions 2290
19-20. DMM Destination x Region 2 (DMMDESTxREG2) Field Descriptions 2291
19-21. DMM Destination x Blocksize 2 (DMMDESTxBL2) Field Descriptions 2292
19-22. DMM Pin Control 0 (DMMPC0) Field Descriptions 2293
19-23. DMM Pin Control 1 (DMMPC1) Field Descriptions 2294
19-24. DMM Pin Control 2 (DMMPC2) Field Descriptions 2296
19-25. DMM Pin Control 3 (DMMPC3) Field Descriptions 2297
19-26. DMM Pin Control 4 (DMMPC4) Field Descriptions 2298
19-27. DMM Pin Control 5 (DMMPC5) Field Descriptions 2300
19-28. DMM Pin Control 6 (DMMPC6) Field Descriptions 2301
19-29. DMM Pin Control 7 (DMMPC7) Field Descriptions 2303
19-30. DMM Pin Control 8 (DMMPC8) Field Descriptions 2304
20-1. ePWM Module Control and Status Register Set Grouped by Submodule 2310
20-2. Submodule Configuration Parameters 2311
20-3. Time-Base Submodule Registers 2313
20-4. Key Time-Base Signals 2314
20-5. Counter-Compare Submodule Registers 2322
20-6. Counter-Compare Submodule Key Signals 2323
20-7. Action-Qualifier Submodule Registers 2326
20-8. Action-Qualifier Submodule Possible Input Events 2327
20-9. Action-Qualifier Event Priority for Up-Down-Count Mode 2329
20-10. Action-Qualifier Event Priority for Up-Count Mode 2329
20-11. Action-Qualifier Event Priority for Down-Count Mode 2329
20-12. Behavior if CMPA/CMPB is Greater than the Period 2329
20-13. Dead-Band Generator Submodule Registers 2339
20-14. Classical Dead-Band Operating Modes 2341
20-15. Dead-Band Delay Values in μS as a Function of DBFED and DBRED 2343
20-16. PWM-Chopper Submodule Registers 2344

20-17. Possible Pulse Width Values for VCLK3 = 100 MHz 2346
20-18. Trip-Zone Submodule Registers 2349
20-19. Possible Actions On a Trip Event 2350
20-20. Event-Trigger Submodule Registers 2356
20-21. Digital Compare Submodule Registers 2360
20-22. MSS_ETPWM1 Registers 2381
20-23. MSS_ETPWM1 Access Type Codes 2382
20-24. TBCTL_TBSTS Register Field Descriptions 2383
20-25. TBPHS Register Field Descriptions 2385
20-26. TBCTR_TBPRD Register Field Descriptions 2386
20-27. CMPCTL Register Field Descriptions 2387
20-28. CMPA Register Field Descriptions 2389
20-29. CMPB_AQCTLA Register Field Descriptions 2390
20-30. AQCTLB_AQSFRC Register Field Descriptions 2392
20-31. AQCSFRC_DBCTL Register Field Descriptions 2394
20-32. DBRED_DBFED Register Field Descriptions 2396
20-33. TZSEL_TZDCSEL Register Field Descriptions 2397
20-34. TZCTL_TZEINT Register Field Descriptions 2399
20-35. TZFLG_TZCLR Register Field Descriptions 2401
20-36. TZFRC_ETSEL Register Field Descriptions 2403
20-37. ETPS_ETFLG Register Field Descriptions 2405
20-38. ETCLR_ETFRC Register Field Descriptions 2407
20-39. PCCTL Register Field Descriptions 2408
20-40. Reserved1 Register Field Descriptions 2409
20-41. Reserved2 Register Field Descriptions 2410
20-42. Reserved3 Register Field Descriptions 2411
20-43. Reserved4 Register Field Descriptions 2412
20-44. Reserved5 Register Field Descriptions 2413
20-45. Reserved6 Register Field Descriptions 2414
20-46. Reserved7 Register Field Descriptions 2415
20-47. Reserved8 Register Field Descriptions 2416
20-48. DCTRIPSEL_DCACTL Register Field Descriptions 2417
20-49. DCBCTL_DCFCTL Register Field Descriptions 2419
20-50. DCCAPCTL_DCFOFFSET Register Field Descriptions 2420
20-51. DCFOFFSETCNT_DCFWINDOW Register Field Descriptions 2421
20-52. DCFWINDOWCNT_DCCAP Register Field Descriptions 2422
21-1. Parameters of the CAN Bit Time 2427
21-2. Message Object Field Descriptions 2434
21-3. Message RAM Addressing in Debug/Suspend and RDA Mode 2436
21-4. Message Interface Register Sets 1 and 2 2439
21-5. Message Interface Register 3 2441
21-6. MSS_DCAN Registers 2461
21-7. MSS_DCAN Access Type Codes 2463
21-8. CTL Register Field Descriptions 2464
21-9. ES Register Field Descriptions 2467
21-10. ERRC Register Field Descriptions 2469
21-11. BTR Register Field Descriptions 2470
21-12. INT Register Field Descriptions 2471
21-13. TEST Register Field Descriptions 2472

21-14. Reserved_1 Register Field Descriptions 2473
21-15. PERR Register Field Descriptions 2474
21-16. DCAN_REV_ID Register Field Descriptions 2475
21-17. ECCDIAG Register Field Descriptions 2476
21-18. ECCDIAG_STAT Register Field Descriptions 2477
21-19. ECC_CS Register Field Descriptions 2478
21-20. ECC_SERR Register Field Descriptions 2479
21-21. TTCAN1 Register Field Descriptions 2480
21-22. TTCAN2 Register Field Descriptions 2481
21-23. TTCAN3 Register Field Descriptions 2482
21-24. TTCAN4 Register Field Descriptions 2483
21-25. TTCAN5 Register Field Descriptions 2484
21-26. TTCAN6 Register Field Descriptions 2485
21-27. TTCAN7 Register Field Descriptions 2486
21-28. TTCAN8 Register Field Descriptions 2487
21-29. TTCAN9 Register Field Descriptions 2488
21-30. TTCAN10 Register Field Descriptions 2489
21-31. TTCAN11 Register Field Descriptions 2490
21-32. TTCAN12 Register Field Descriptions 2491
21-33. TTCAN13 Register Field Descriptions 2492
21-34. TTCAN14 Register Field Descriptions 2493
21-35. TTCAN15 Register Field Descriptions 2494
21-36. TTCAN16 Register Field Descriptions 2495
21-37. TTCAN17 Register Field Descriptions 2496
21-38. TTCAN18 Register Field Descriptions 2497
21-39. TTCAN19 Register Field Descriptions 2498
21-40. ABOTR Register Field Descriptions 2499
21-41. TXRQ_X Register Field Descriptions 2500
21-42. TXRQ12 Register Field Descriptions 2501
21-43. TXRQ34 Register Field Descriptions 2502
21-44. TXRQ56 Register Field Descriptions 2503
21-45. TXRQ78 Register Field Descriptions 2504
21-46. NWDAT_X Register Field Descriptions 2505
21-47. NWDAT12 Register Field Descriptions 2506
21-48. NWDAT34 Register Field Descriptions 2507
21-49. NWDAT56 Register Field Descriptions 2508
21-50. NWDAT78 Register Field Descriptions 2509
21-51. INTPND_X Register Field Descriptions 2510
21-52. INTPND12 Register Field Descriptions 2511
21-53. INTPND34 Register Field Descriptions 2512
21-54. INTPND56 Register Field Descriptions 2513
21-55. INTPND78 Register Field Descriptions 2514
21-56. MSGVAL_X Register Field Descriptions 2515
21-57. MSGVAL12 Register Field Descriptions 2516
21-58. MSGVAL34 Register Field Descriptions 2517
21-59. MSGVAL56 Register Field Descriptions 2518
21-60. MSGVAL78 Register Field Descriptions 2519
21-61. Reserved_2 Register Field Descriptions 2520
21-62. INTMUX12 Register Field Descriptions 2521

21-63. INTMUX34 Register Field Descriptions 2522
21-64. INTMUX56 Register Field Descriptions 2523
21-65. INTMUX78 Register Field Descriptions 2524
21-66. Reserved_3 Register Field Descriptions 2525
21-67. Reserved_4 Register Field Descriptions 2526
21-68. Reserved_5 Register Field Descriptions 2527
21-69. Reserved_6 Register Field Descriptions 2528
21-70. Reserved_7 Register Field Descriptions 2529
21-71. Reserved_8 Register Field Descriptions 2530
21-72. IF1CMD Register Field Descriptions 2531
21-73. IF1MSK Register Field Descriptions 2534
21-74. IF1ARB Register Field Descriptions 2535
21-75. IF1MCTL Register Field Descriptions 2536
21-76. IF1DATA Register Field Descriptions 2538
21-77. IF1DATB Register Field Descriptions 2539
21-78. Reserved_9 Register Field Descriptions 2540
21-79. Reserved_10 Register Field Descriptions 2541
21-80. IF2CMD Register Field Descriptions 2542
21-81. IF2MSK Register Field Descriptions 2545
21-82. IF2ARB Register Field Descriptions 2546
21-83. IF2MCTL Register Field Descriptions 2547
21-84. IF2DATA Register Field Descriptions 2549
21-85. IF2DATB Register Field Descriptions 2550
21-86. Reserved_11 Register Field Descriptions 2551
21-87. Reserved_12 Register Field Descriptions 2552
21-88. IF3OBS Register Field Descriptions 2553
21-89. IF3MSK Register Field Descriptions 2555
21-90. IF3ARB Register Field Descriptions 2556
21-91. IF3MCTL Register Field Descriptions 2557
21-92. IF3DATA Register Field Descriptions 2559
21-93. IF3DATB Register Field Descriptions 2560
21-94. Reserved_13 Register Field Descriptions 2561
21-95. Reserved_14 Register Field Descriptions 2562
21-96. IF3UPD12 Register Field Descriptions 2563
21-97. IF3UPD34 Register Field Descriptions 2564
21-98. IF3UPD56 Register Field Descriptions 2565
21-99. IF3UPD78 Register Field Descriptions 2566
21-100. Reserved_15 Register Field Descriptions 2567
21-101. Reserved_16 Register Field Descriptions 2568
21-102. Reserved_17 Register Field Descriptions 2569
21-103. Reserved_18 Register Field Descriptions 2570
21-104. Reserved_19 Register Field Descriptions 2571
21-105. Reserved_20 Register Field Descriptions 2572
21-106. Reserved_21 Register Field Descriptions 2573
21-107. Reserved_22 Register Field Descriptions 2574
21-108. Reserved_23 Register Field Descriptions 2575
21-109. Reserved_24 Register Field Descriptions 2576
21-110. Reserved_25 Register Field Descriptions 2577
21-111. Reserved_26 Register Field Descriptions 2578

21-112. Reserved_27 Register Field Descriptions 2579
21-113. Reserved_28 Register Field Descriptions 2580
21-114. Reserved_29 Register Field Descriptions 2581
21-115. Reserved_30 Register Field Descriptions 2582
21-116. Reserved_31 Register Field Descriptions 2583
21-117. Reserved_32 Register Field Descriptions 2584
21-118. Reserved_33 Register Field Descriptions 2585
21-119. Reserved_34 Register Field Descriptions 2586
21-120. Reserved_35 Register Field Descriptions 2587
21-121. Reserved_36 Register Field Descriptions 2588
21-122. Reserved_37 Register Field Descriptions 2589
21-123. Reserved_38 Register Field Descriptions 2590
21-124. Reserved_39 Register Field Descriptions 2591
21-125. Reserved_40 Register Field Descriptions 2592
21-126. Reserved_41 Register Field Descriptions 2593
21-127. Reserved_42 Register Field Descriptions 2594
21-128. TIOC Register Field Descriptions 2595
21-129. RIOC Register Field Descriptions 2597
22-1. DLC Coding in CAN FD 2605
22-2. Rx Buffer / Rx FIFO Element Size 2618
22-3. Example Filter Configuration for Rx Buffers 2619
22-4. Possible Configurations for Message Transmission 2620
22-5. Tx Buffer / Tx FIFO / Tx Queue Element Size 2621
22-6. Rx Buffer / Rx FIFO Element Field Descriptions 2626
22-7. Tx Buffer Element Field Descriptions 2628
22-8. Tx Event FIFO Element Field Descriptions 2629
22-9. Standard Message ID Filter Element Field Descriptions 2631
22-10. Extended Message ID Filter Element Field Descriptions 2632
22-11. MSS_MCAN_CFG Registers 2635
22-12. MSS_MCAN_CFG Access Type Codes 2636
22-13. MCANSS_PID Register Field Descriptions 2637
22-14. MCANSS_CTRL Register Field Descriptions 2638
22-15. MCANSS_STAT Register Field Descriptions 2639
22-16. MCANSS_ICS Register Field Descriptions 2640
22-17. MCANSS_IRS Register Field Descriptions 2641
22-18. MCANSS_IECS Register Field Descriptions 2642
22-19. MCANSS_IE Register Field Descriptions 2643
22-20. MCANSS_IES Register Field Descriptions 2644
22-21. MCANSS_EOI Register Field Descriptions 2645
22-22. MCANSS_EXT_TS_PS Register Field Descriptions 2646
22-23. MCANSS_EXT_TS_USIC Register Field Descriptions 2647
22-24. MCANSS_ECC_EOI Register Field Descriptions 2648
22-25. MCAN_CREL Register Field Descriptions 2649
22-26. MCAN_ENDN Register Field Descriptions 2650
22-27. MCAN_CUST Register Field Descriptions 2651
22-28. MCAN_DBTP Register Field Descriptions 2652
22-29. MCAN_TEST Register Field Descriptions 2653
22-30. MCAN_RWD Register Field Descriptions 2654
22-31. MCAN_CCCR Register Field Descriptions 2655

22-32. MCAN_NBTP Register Field Descriptions 2656
22-33. MCAN_TSCC Register Field Descriptions 2657
22-34. MCAN_TSCV Register Field Descriptions 2658
22-35. MCAN_TOCC Register Field Descriptions 2659
22-36. MCAN_TOCV Register Field Descriptions 2660
22-37. MCAN_RES00 Register Field Descriptions 2661
22-38. MCAN_RES01 Register Field Descriptions 2662
22-39. MCAN_RES02 Register Field Descriptions 2663
22-40. MCAN_RES03 Register Field Descriptions 2664
22-41. MCAN_ECR Register Field Descriptions 2665
22-42. MCAN_PSR Register Field Descriptions 2666
22-43. MCAN_TDCR Register Field Descriptions 2667
22-44. MCAN_RES04 Register Field Descriptions 2668
22-45. MCAN_IR Register Field Descriptions 2669
22-46. MCAN_IE Register Field Descriptions 2671
22-47. MCAN_ILS Register Field Descriptions 2673
22-48. MCAN_ILE Register Field Descriptions 2675
22-49. MCAN_RES05 Register Field Descriptions 2676
22-50. MCAN_RES06 Register Field Descriptions 2677
22-51. MCAN_RES07 Register Field Descriptions 2678
22-52. MCAN_RES08 Register Field Descriptions 2679
22-53. MCAN_RES09 Register Field Descriptions 2680
22-54. MCAN_RES10 Register Field Descriptions 2681
22-55. MCAN_RES11 Register Field Descriptions 2682
22-56. MCAN_RES12 Register Field Descriptions 2683
22-57. MCAN_GFC Register Field Descriptions 2684
22-58. MCAN_SIDFC Register Field Descriptions 2685
22-59. MCAN_XIDFC Register Field Descriptions 2686
22-60. MCAN_RES13 Register Field Descriptions 2687
22-61. MCAN_XIDAM Register Field Descriptions 2688
22-62. MCAN_HPMS Register Field Descriptions 2689
22-63. MCAN_NDAT1 Register Field Descriptions 2690
22-64. MCAN_NDAT2 Register Field Descriptions 2691
22-65. MCAN_RXF0C Register Field Descriptions 2692
22-66. MCAN_RXF0S Register Field Descriptions 2693
22-67. MCAN_RXF0A Register Field Descriptions 2694
22-68. MCAN_RXBC Register Field Descriptions 2695
22-69. MCAN_RXF1C Register Field Descriptions 2696
22-70. MCAN_RXF1S Register Field Descriptions 2697
22-71. MCAN_RXF1A Register Field Descriptions 2698
22-72. MCAN_RXESC Register Field Descriptions 2699
22-73. MCAN_TXBC Register Field Descriptions 2700
22-74. MCAN_TXFQS Register Field Descriptions 2701
22-75. MCAN_TXESC Register Field Descriptions 2702
22-76. MCAN_TXBRP Register Field Descriptions 2703
22-77. MCAN_TXBAR Register Field Descriptions 2704
22-78. MCAN_TXBCR Register Field Descriptions 2705
22-79. MCAN_TXBTO Register Field Descriptions 2706
22-80. MCAN_TXBCF Register Field Descriptions 2707

22-81. MCAN_TXBTIE Register Field Descriptions 2708
22-82. MCAN_TXBCIE Register Field Descriptions 2709
22-83. MCAN_RES14 Register Field Descriptions 2710
22-84. MCAN_RES15 Register Field Descriptions 2711
22-85. MCAN_TXEFC Register Field Descriptions 2712
22-86. MCAN_TXEFS Register Field Descriptions 2713
22-87. MCAN_TXEFA Register Field Descriptions 2714
22-88. MCAN_RES16 Register Field Descriptions 2715
22-89. MSS_MCAN_ECC Registers 2716
22-90. MSS_MCAN_ECC Access Type Codes 2716
22-91. MCANSS_ECC_AGGR_REVISION Register Field Descriptions 2717
22-92. MCANSS_ECC_VECTOR Register Field Descriptions 2718
22-93. MCANSS_ECC_MISC_STATUS Register Field Descriptions 2719
22-94. MCANSS_ECC_WRAP_REVISION Register Field Descriptions 2720
22-95. MCANSS_ECC_CONTROL Register Field Descriptions 2721
22-96. MCANSS_ECC_ERR_CTRL1 Register Field Descriptions 2722
22-97. MCANSS_ECC_ERR_CTRL2 Register Field Descriptions 2723
22-98. MCANSS_ECC_ERR_STAT1 Register Field Descriptions 2724
22-99. MCANSS_ECC_ERR_STAT2 Register Field Descriptions 2725
22-100. MCANSS_ECC_SEC_EOI_REG Register Field Descriptions 2726
22-101. MCANSS_ECC_SEC_STATUS_REG0 Register Field Descriptions 2727
22-102. MCANSS_ECC_SEC_ENABLE_SET_REG0 Register Field Descriptions 2728
22-103. MCANSS_ECC_SEC_ENABLE_CLR_REG0 Register Field Descriptions 2729
22-104. MCANSS_ECC_DED_EOI_REG Register Field Descriptions 2730
22-105. MCANSS_ECC_DED_STATUS_REG0 Register Field Descriptions 2731
22-106. MCANSS_ECC_DED_ENABLE_SET_REG0 Register Field Descriptions 2732
22-107. MCANSS_ECC_DED_ENABLE_CLR_REG0 Register Field Descriptions 2733
22-108. DLC Coding in CAN FD 2739
22-109. Rx Buffer / Rx FIFO Element Size 2752
22-110. Example Filter Configuration for Rx Buffers 2753
22-111. Possible Configurations for Message Transmission 2754
22-112. Tx Buffer / Tx FIFO / Tx Queue Element Size 2755
22-113. Rx Buffer / Rx FIFO Element Field Descriptions 2760
22-114. Tx Buffer Element Field Descriptions 2762
22-115. Tx Event FIFO Element Field Descriptions 2763
22-116. Standard Message ID Filter Element Field Descriptions 2765
22-117. Extended Message ID Filter Element Field Descriptions 2766
22-118. MSS_MCAN_CFG Registers 2769
22-119. SS_PID Register Field Descriptions 2771
22-120. SS_CTRL Register Field Descriptions 2772
22-121. SS_STAT Register Field Descriptions 2773
22-122. SS_ICS Register Field Descriptions 2774
22-123. SS_IRS Register Field Descriptions 2775
22-124. SS_IECS Register Field Descriptions 2776
22-125. SS_IE Register Field Descriptions 2777
22-126. SS_IES Register Field Descriptions 2778
22-127. SS_EOI Register Field Descriptions 2779
22-128. SS_EXT_TS_PS Register Field Descriptions 2780
22-129. SS_EXT_TS_USIC Register Field Descriptions 2781

22-130. CREL Register Field Descriptions 2782
22-131. ENDN Register Field Descriptions 2783
22-132. CUST Register Field Descriptions 2784
22-133. DBTP Register Field Descriptions 2785
22-134. TEST Register Field Descriptions 2786
22-135. RWD Register Field Descriptions 2787
22-136. CCCR Register Field Descriptions 2788
22-137. NBTP Register Field Descriptions 2789
22-138. TSCC Register Field Descriptions 2790
22-139. TSCV Register Field Descriptions 2791
22-140. TOCC Register Field Descriptions 2792
22-141. TOCV Register Field Descriptions 2793
22-142. RES00 Register Field Descriptions 2794
22-143. RES01 Register Field Descriptions 2795
22-144. RES02 Register Field Descriptions 2796
22-145. RES03 Register Field Descriptions 2797
22-146. ECR Register Field Descriptions 2798
22-147. PSR Register Field Descriptions 2799
22-148. TDCR Register Field Descriptions 2800
22-149. RES04 Register Field Descriptions 2801
22-150. IR Register Field Descriptions 2802
22-151. IE Register Field Descriptions 2804
22-152. ILS Register Field Descriptions 2806
22-153. ILE Register Field Descriptions 2808
22-154. RES05 Register Field Descriptions 2809
22-155. RES06 Register Field Descriptions 2810
22-156. RES07 Register Field Descriptions 2811
22-157. RES08 Register Field Descriptions 2812
22-158. RES09 Register Field Descriptions 2813
22-159. RES10 Register Field Descriptions 2814
22-160. RES11 Register Field Descriptions 2815
22-161. RES12 Register Field Descriptions 2816
22-162. GFC Register Field Descriptions 2817
22-163. SIDFC Register Field Descriptions 2818
22-164. XIDFC Register Field Descriptions 2819
22-165. RES13 Register Field Descriptions 2820
22-166. XIDAM Register Field Descriptions 2821
22-167. HPMS Register Field Descriptions 2822
22-168. NDAT1 Register Field Descriptions 2823
22-169. NDAT2 Register Field Descriptions 2824
22-170. RXF0C Register Field Descriptions 2825
22-171. RXF0S Register Field Descriptions 2826
22-172. RXF0A Register Field Descriptions 2827
22-173. RXBC Register Field Descriptions 2828
22-174. RXF1C Register Field Descriptions 2829
22-175. RXF1S Register Field Descriptions 2830
22-176. RXF1A Register Field Descriptions 2831
22-177. RXESC Register Field Descriptions 2832
22-178. TXBC Register Field Descriptions 2833

22-179. TXFQS Register Field Descriptions 2834
22-180. TXESC Register Field Descriptions 2835
22-181. TXBRP Register Field Descriptions 2836
22-182. TXBAR Register Field Descriptions 2837
22-183. TXBCR Register Field Descriptions 2838
22-184. TXBTO Register Field Descriptions 2839
22-185. TXBCF Register Field Descriptions 2840
22-186. TXBTIE Register Field Descriptions 2841
22-187. TXBCIE Register Field Descriptions 2842
22-188. RES14 Register Field Descriptions 2843
22-189. RES15 Register Field Descriptions 2844
22-190. TXEFC Register Field Descriptions 2845
22-191. TXEFS Register Field Descriptions 2846
22-192. TXEFA Register Field Descriptions 2847
22-193. RES16 Register Field Descriptions 2848
22-194. MSS_MCAN_ECC Registers 2849
22-195. REV Register Field Descriptions 2850
22-196. VECTOR Register Field Descriptions 2851
22-197. STAT Register Field Descriptions 2852
22-198. CTRL Register Field Descriptions 2853
22-199. ERR_CTRL1 Register Field Descriptions 2854
22-200. ERR_CTRL2 Register Field Descriptions 2855
22-201. ERR_STAT1 Register Field Descriptions 2856
22-202. ERR_STAT2 Register Field Descriptions 2857
22-203. ERR_STAT3 Register Field Descriptions 2858
22-204. SEC_EOI_REG Register Field Descriptions 2859
22-205. SEC_STATUS_REG0 Register Field Descriptions 2860
22-206. SEC_ENABLE_SET_REG0 Register Field Descriptions 2861
22-207. SEC_ENABLE_CLR_REG0 Register Field Descriptions 2862
22-208. DED_EOI_REG Register Field Descriptions 2863
22-209. DED_STATUS_REG0 Register Field Descriptions 2864
22-210. DED_ENABLE_SET_REG0 Register Field Descriptions 2865
22-211. DED_ENABLE_CLR_REG0 Register Field Descriptions 2866
22-212. AGGR_ENABLE_SET Register Field Descriptions 2867
22-213. AGGR_ENABLE_CLR Register Field Descriptions 2868
22-214. AGGR_STATUS_SET Register Field Descriptions 2869
22-215. AGGR_STATUS_CLR Register Field Descriptions 2870
23-1. Register Control From MSS-RCM Configuration Space 2871
23-2. Register Control From MSS-RCM Configuration Space 2871
23-3. Pin Configurations 2873
23-4. MibSPI/SPI Configurations 2874
23-5. Clocking Modes 2886
23-6. Pin Mapping for SIMO Pin with MSB First 2895
23-7. Pin Mapping for SOMI Pin with MSB First 2895
23-8. Pin Mapping for SIMO Pin with LSB First 2896
23-9. Pin Mapping for SOMI Pin with LSB First 2896
23-10. MSS_MIBSPIA Registers 2910
23-11. MSS_MIBSPIA Access Type Codes 2912
23-12. SPIGCR0 Register Field Descriptions 2913

23-13. SPIGCR1 Register Field Descriptions 2914
23-14. SPIINT0 Register Field Descriptions 2916
23-15. SPILVL Register Field Descriptions 2918
23-16. SPIFLG Register Field Descriptions 2919
23-17. SPIPC0 Register Field Descriptions 2922
23-18. SPIPC1 Register Field Descriptions 2924
23-19. SPIPC2 Register Field Descriptions 2926
23-20. SPIPC3 Register Field Descriptions 2927
23-21. SPIPC4 Register Field Descriptions 2929
23-22. SPIPC5 Register Field Descriptions 2931
23-23. SPIPC6 Register Field Descriptions 2933
23-24. SPIDAT0 Register Field Descriptions 2935
23-25. SPIDAT1 Register Field Descriptions 2936
23-26. SPIBUF Register Field Descriptions 2938
23-27. SPIEMU Register Field Descriptions 2941
23-28. SPIDELAY Register Field Descriptions 2943
23-29. SPIDEF Register Field Descriptions 2945
23-30. SPIFMT0 Register Field Descriptions 2946
23-31. SPIFMT1 Register Field Descriptions 2948
23-32. SPIFMT2 Register Field Descriptions 2950
23-33. SPIFMT3 Register Field Descriptions 2952
23-34. TGINTVECT0 Register Field Descriptions 2954
23-35. TGINTVECT1 Register Field Descriptions 2956
23-36. SPIPC9 Register Field Descriptions 2958
23-37. SPIPMCTRL Register Field Descriptions 2959
23-38. MIBSPIE Register Field Descriptions 2962
23-39. TGITENST Register Field Descriptions 2964
23-40. TGITENCR Register Field Descriptions 2965
23-41. TGITLVST Register Field Descriptions 2966
23-42. TGITLVCR Register Field Descriptions 2967
23-43. TGINTFLAG Register Field Descriptions 2968
23-44. TICKCNT Register Field Descriptions 2969
23-45. LTGPEND Register Field Descriptions 2970
23-46. TG0CTRL Register Field Descriptions 2971
23-47. TG1CTRL Register Field Descriptions 2974
23-48. TG2CTRL Register Field Descriptions 2977
23-49. TG3CTRL Register Field Descriptions 2980
23-50. TG4CTRL Register Field Descriptions 2983
23-51. TG5CTRL Register Field Descriptions 2986
23-52. TG6CTRL Register Field Descriptions 2989
23-53. TG7CTRL Register Field Descriptions 2992
23-54. DMA0CTRL Register Field Descriptions 2995
23-55. DMA1CTRL Register Field Descriptions 2997
23-56. DMA2CTRL Register Field Descriptions 2999
23-57. DMA3CTRL Register Field Descriptions 3001
23-58. DMA4CTRL Register Field Descriptions 3003
23-59. ICOUNT0 Register Field Descriptions 3005
23-60. ICOUNT1 Register Field Descriptions 3006
23-61. ICOUNT2 Register Field Descriptions 3007

23-62. ICOUNT3 Register Field Descriptions 3008
23-63. ICOUNT4 Register Field Descriptions 3009
23-64. DMACNTLEN Register Field Descriptions 3010
23-65. PAR_ECC_CTRL Register Field Descriptions 3011
23-66. PAR_ECC_STAT Register Field Descriptions 3012
23-67. UERRADDR1 Register Field Descriptions 3013
23-68. UERRADDR0 Register Field Descriptions 3014
23-69. RXOVRN_BUF_ADDR Register Field Descriptions 3015
23-70. IOLPBKTSTCR Register Field Descriptions 3016
23-71. EXTENDED_PRESCALE1 Register Field Descriptions 3018
23-72. EXTENDED_PRESCALE2 Register Field Descriptions 3020
23-73. ECCDIAG_CTRL Register Field Descriptions 3021
23-74. ECCDIAG_STAT Register Field Descriptions 3022
23-75. SBERRADDR1 Register Field Descriptions 3023
23-76. SBERRADDR0 Register Field Descriptions 3024
23-77. SPIREV Register Field Descriptions 3025
23-78. MSS_MIBSPIB Registers 3026
23-79. MSS_MIBSPIB Access Type Codes 3028
23-80. SPIGCR0 Register Field Descriptions 3029
23-81. SPIGCR1 Register Field Descriptions 3030
23-82. SPIINT0 Register Field Descriptions 3032
23-83. SPILVL Register Field Descriptions 3034
23-84. SPIFLG Register Field Descriptions 3035
23-85. SPIPC0 Register Field Descriptions 3038
23-86. SPIPC1 Register Field Descriptions 3040
23-87. SPIPC2 Register Field Descriptions 3042
23-88. SPIPC3 Register Field Descriptions 3043
23-89. SPIPC4 Register Field Descriptions 3045
23-90. SPIPC5 Register Field Descriptions 3047
23-91. SPIPC6 Register Field Descriptions 3049
23-92. SPIDAT0 Register Field Descriptions 3051
23-93. SPIDAT1 Register Field Descriptions 3052
23-94. SPIBUF Register Field Descriptions 3054
23-95. SPIEMU Register Field Descriptions 3057
23-96. SPIDELAY Register Field Descriptions 3059
23-97. SPIDEF Register Field Descriptions 3061
23-98. SPIFMT0 Register Field Descriptions 3062
23-99. SPIFMT1 Register Field Descriptions 3064
23-100. SPIFMT2 Register Field Descriptions 3066
23-101. SPIFMT3 Register Field Descriptions 3068
23-102. TGINTVECT0 Register Field Descriptions 3070
23-103. TGINTVECT1 Register Field Descriptions 3072
23-104. SPIPC9 Register Field Descriptions 3074
23-105. SPIPMCTRL Register Field Descriptions 3075
23-106. MIBSPIE Register Field Descriptions 3078
23-107. TGITENST Register Field Descriptions 3080
23-108. TGITENCR Register Field Descriptions 3081
23-109. TGITLVST Register Field Descriptions 3082
23-110. TGITLVCR Register Field Descriptions 3083

23-111. TGINTFLAG Register Field Descriptions 3084
23-112. TICKCNT Register Field Descriptions 3085
23-113. LTGPEND Register Field Descriptions 3086
23-114. TG0CTRL Register Field Descriptions 3087
23-115. TG1CTRL Register Field Descriptions 3090
23-116. TG2CTRL Register Field Descriptions 3093
23-117. TG3CTRL Register Field Descriptions 3096
23-118. TG4CTRL Register Field Descriptions 3099
23-119. TG5CTRL Register Field Descriptions 3102
23-120. TG6CTRL Register Field Descriptions 3105
23-121. TG7CTRL Register Field Descriptions 3108
23-122. DMA0CTRL Register Field Descriptions 3111
23-123. DMA1CTRL Register Field Descriptions 3113
23-124. DMA2CTRL Register Field Descriptions 3115
23-125. DMA3CTRL Register Field Descriptions 3117
23-126. DMA4CTRL Register Field Descriptions 3119
23-127. ICOUNT0 Register Field Descriptions 3121
23-128. ICOUNT1 Register Field Descriptions 3122
23-129. ICOUNT2 Register Field Descriptions 3123
23-130. ICOUNT3 Register Field Descriptions 3124
23-131. ICOUNT4 Register Field Descriptions 3125
23-132. DMACNTLEN Register Field Descriptions 3126
23-133. PAR_ECC_CTRL Register Field Descriptions 3127
23-134. PAR_ECC_STAT Register Field Descriptions 3128
23-135. UERRADDR1 Register Field Descriptions 3129
23-136. UERRADDR0 Register Field Descriptions 3130
23-137. RXOVRN_BUF_ADDR Register Field Descriptions 3131
23-138. IOLPBKTSTCR Register Field Descriptions 3132
23-139. EXTENDED_PRESCALE1 Register Field Descriptions 3134
23-140. EXTENDED_PRESCALE2 Register Field Descriptions 3136
23-141. ECCDIAG_CTRL Register Field Descriptions 3137
23-142. ECCDIAG_STAT Register Field Descriptions 3138
23-143. SBERRADDR1 Register Field Descriptions 3139
23-144. SBERRADDR0 Register Field Descriptions 3140
23-145. SPIREV Register Field Descriptions 3141
23-146. Multi-Buffer RAM Register 3143
23-147. Multi-buffer RAM Transmit Data Register Field Descriptions 3144
23-148. Multi-buffer Receive Buffer Register Field Descriptions 3146
24-1. SPI Clock Modes Definition 3163
24-2. QSPI Events 3166
24-3. MSS_QSPI Registers 3168
24-4. MSS_QSPI Access Type Codes 3168
24-5. PID Register Field Descriptions 3169
24-6. SYSCONFIG Register Field Descriptions 3170
24-7. INTR_STATUS_RAW_SET Register Field Descriptions 3171
24-8. INTR_STATUS_ENABLED_CLEAR Register Field Descriptions 3172
24-9. INTR_ENABLE_SET Register Field Descriptions 3173
24-10. INTR_ENABLE_CLEAR Register Field Descriptions 3174
24-11. INTC_EOI Register Field Descriptions 3175

24-12. SPI_CLOCK_CNTRL Register Field Descriptions 3176
24-13. SPI_DC Register Field Descriptions 3177
24-14. SPI_CMD Register Field Descriptions 3179
24-15. SPI_STATUS Register Field Descriptions 3180
24-16. SPI_DATA Register Field Descriptions 3181
24-17. SPI_SETUP0 Register Field Descriptions 3182
24-18. SPI_SETUP1 Register Field Descriptions 3183
24-19. SPI_SETUP2 Register Field Descriptions 3184
24-20. SPI_SETUP3 Register Field Descriptions 3185
24-21. SPI_SWITCH Register Field Descriptions 3186
24-22. SPI_DATA1 Register Field Descriptions 3187
24-23. SPI_DATA2 Register Field Descriptions 3188
24-24. SPI_DATA3 Register Field Descriptions 3189
25-1. Ways to Generate a NACK Bit 3198
25-2. Interrupt Requests Generated by I2C Module 3203
25-3. MSS_I2C Registers 3206
25-4. MSS_I2C Access Type Codes 3206
25-5. ICOAR Register Field Descriptions 3207
25-6. ICIMR Register Field Descriptions 3208
25-7. ICSTR Register Field Descriptions 3209
25-8. ICCLKL Register Field Descriptions 3211
25-9. ICCLKH Register Field Descriptions 3212
25-10. ICCNT Register Field Descriptions 3213
25-11. ICDRR Register Field Descriptions 3214
25-12. ICSAR Register Field Descriptions 3215
25-13. ICDXR Register Field Descriptions 3216
25-14. ICMDR Register Field Descriptions 3217
25-15. ICIVR Register Field Descriptions 3220
25-16. ICEMDR Register Field Descriptions 3221
25-17. ICPSC Register Field Descriptions 3222
25-18. ICPID1 Register Field Descriptions 3223
25-19. ICPID2 Register Field Descriptions 3224
25-20. ICDMAC Register Field Descriptions 3225
25-21. ICPFUNC Register Field Descriptions 3226
25-22. ICPDIR Register Field Descriptions 3227
25-23. ICPDIN Register Field Descriptions 3228
25-24. ICPDOUT Register Field Descriptions 3229
25-25. ICPDSET Register Field Descriptions 3230
25-26. ICPDCLR Register Field Descriptions 3231
25-27. ICPDRV Register Field Descriptions 3232
26-1. SCI Interrupts 3244
26-2. DMA and Interrupt Requests in Multiprocessor Modes 3245
26-3. SCI Control Registers Summary 3250
26-4. SCI Global Control Register 0 (SCIGCR0) Fied Descriptions 3251
26-5. SCI Global Control Register 1 (SCIGCR1) Field Descriptions 3252
26-6. SCI Set Interrupt Register (SCISETINT) Field Descriptions 3255
26-7. SCI Clear Interrupt Register (SCICLEARINT) Field Descriptions 3257
26-8. SCI Set Interrupt Level Register (SCISETINTLVL) Field Descriptions 3259
26-9. SCI Clear Interrupt Level Register (SCICLEARINTLVL) Field Descriptions 3260

26-10. SCI Flags Register (SCIFLR) Field Descriptions 3262
26-11. SCI Receiver Status Flags 3265
26-12. SCI Transmitter Status Flags 3265
26-13. SCI Interrupt Vector Offset 0 (SCIINTVECT0) Field Descriptions 3266
26-14. SCI Interrupt Vector Offset 1 (SCIINTVECT1) Field Descriptions 3266
26-15. SCI Format Control Register (SCIFORMAT) Field Descriptions 3267
26-16. Baud Rate Selection Register (BRS) Field Descriptions 3268
26-17. Comparative Baud Values for Different P Values, Asynchronous Mode 3268
26-18. Receiver Emulation Data Buffer (SCIED) Field Descriptions 3269
26-19. Receiver Data Buffer (SCIRD) Field Descriptions 3269
26-20. Transmit Data Buffer Register (SCITD) Field Descriptions 3270
26-21. SCI Pin I/O Control Register 0 (SCIPIO0) Field Descriptions 3270
26-22. SCI Pin I/O Control Register 1 (SCIPIO1) Field Descriptions 3271
26-23. SCITX Pin Control 3271
26-24. SCIRX Pin Control 3271
26-25. SCI Pin I/O Control Register 2 (SCIPIO2) Field Descriptions 3272
26-26. SCI Pin I/O Control Register 3 (SCIPIO3) Field Descriptions 3273
26-27. SCI Pin I/O Control Register 4 (SCIPIO4) Field Descriptions 3274
26-28. SCI Pin I/O Control Register 5 (SCIPIO5) Field Descriptions 3275
26-29. SCI Pin I/O Control Register 6 (SCIPIO6) Field Descriptions 3276
26-30. SCI Pin I/O Control Register 7 (SCIPIO7) Field Descriptions 3277
26-31. SCI Pin I/O Control Register 8 (SCIPIO8) Field Descriptions 3277
26-32. Input/Output Error Enable Register (IODFTCTRL) Field Descriptions 3278
26-33. Input Buffer, Output Buffer, and Pull Control Behavior as GPIO Pins 3281
27-1. 14xx Debug Subsystem Address Map 3283
27-2. 16xx/18xx/68xx Debug Subsystem Address Map 3283
27-3. MSS CTI Trigger Inputs 3284
27-4. MSS CTI Trigger Outputs 3284
27-5. Debug SS CTI Trigger Inputs 3284
28-1. DCC Registers 3292
28-2. DCC Access Type Codes 3292
28-3. DCCGCTRL Register Field Descriptions 3293
28-4. DCCREV Register Field Descriptions 3294
28-5. DCCCNTSEED0 Register Field Descriptions 3295
28-6. DCCVALIDSEED0 Register Field Descriptions 3296
28-7. DCCCNTSEED1 Register Field Descriptions 3297
28-8. DCCSTAT Register Field Descriptions 3298
28-9. DCCCNT0 Register Field Descriptions 3299
28-10. DCCVALID0 Register Field Descriptions 3300
28-11. DCCCNT1 Register Field Descriptions 3301
28-12. DCCCLKSSRC1 Register Field Descriptions 3302
28-13. DCCCLKSSRC0 Register Field Descriptions 3303
28-14. ESM Interrupt and $\overline{\text{ERROR}}$ Pin Behavior 3305
28-15. MSS_ESM Registers 3310
28-16. MSS_ESM Access Type Codes 3310
28-17. ESMIEPSR1 Register Field Descriptions 3311
28-18. ESMIEPCR1 Register Field Descriptions 3312
28-19. ESMIESR1 Register Field Descriptions 3313
28-20. ESMIECR1 Register Field Descriptions 3314

28-21. ESMILSR1 Register Field Descriptions 3315
28-22. ESMILCR1 Register Field Descriptions 3316
28-23. ESMSR1 Register Field Descriptions 3317
28-24. ESMSR2 Register Field Descriptions 3318
28-25. ESMSR3 Register Field Descriptions 3319
28-26. ESMEPSR Register Field Descriptions 3320
28-27. ESMIOFFHR Register Field Descriptions 3321
28-28. ESMIOFFLR Register Field Descriptions 3322
28-29. ESMLTCR Register Field Descriptions 3323
28-30. ESMLTCPR Register Field Descriptions 3324
28-31. ESMEKR Register Field Descriptions 3325
28-32. ESMSSR2 Register Field Descriptions 3326
28-33. ESMIEPSR4 Register Field Descriptions 3327
28-34. ESMIEPCR4 Register Field Descriptions 3328
28-35. ESMIESR4 Register Field Descriptions 3329
28-36. ESMIECR4 Register Field Descriptions 3330
28-37. ESMILSR4 Register Field Descriptions 3331
28-38. ESMILCR4 Register Field Descriptions 3332
28-39. ESMSR4 Register Field Descriptions 3333
28-40. ESMIEPSR7 Register Field Descriptions 3334
28-41. ESMIEPCR7 Register Field Descriptions 3335
28-42. ESMIESR7 Register Field Descriptions 3336
28-43. ESMIECR7 Register Field Descriptions 3337
28-44. ESMILSR7 Register Field Descriptions 3338
28-45. ESMILCR7 Register Field Descriptions 3339
28-46. ESMSR7 Register Field Descriptions 3340
28-47. CRC Modes in Which DMA Request and Counter Logic are Active or Inactive 3347
28-48. Modes in Which Interrupt Condition Can Occur 3348
28-49. Interrupt Offset Mapping 3351
28-50. MSS_MCRC Registers 3356
28-51. MSS_MCRC Access Type Codes 3357
28-52. CRC_CTRL0 Register Field Descriptions 3358
28-53. CRC_CTRL1 Register Field Descriptions 3360
28-54. CRC_CTRL2 Register Field Descriptions 3361
28-55. CRC_INTS Register Field Descriptions 3362
28-56. CRC_INTR Register Field Descriptions 3365
28-57. CRC_STATUS_REG Register Field Descriptions 3368
28-58. CRC_INT_OFFSET_REG Register Field Descriptions 3370
28-59. CRC_BUSY Register Field Descriptions 3371
28-60. CRC_PCOUNT_REG1 Register Field Descriptions 3372
28-61. CRC_SCOUNT_REG1 Register Field Descriptions 3373
28-62. CRC_CURSEC_REG1 Register Field Descriptions 3374
28-63. CRC_WDTOPLD1 Register Field Descriptions 3375
28-64. CRC_BCTOPLD1 Register Field Descriptions 3376
28-65. PSA_SIGREGL1 Register Field Descriptions 3377
28-66. PSA_SIGREGH1 Register Field Descriptions 3378
28-67. CRC_REGL1 Register Field Descriptions 3379
28-68. CRC_REGH1 Register Field Descriptions 3380
28-69. PSA_SECSIGREGL1 Register Field Descriptions 3381

28-70. PSA_SECSIGREGH1 Register Field Descriptions 3382
28-71. RAW_DATAREGL1 Register Field Descriptions 3383
28-72. RAW_DATAREGH1 Register Field Descriptions 3384
28-73. CRC_PCOUNT_REG2 Register Field Descriptions 3385
28-74. CRC_SCOUNT_REG2 Register Field Descriptions 3386
28-75. CRC_CURSEC_REG2 Register Field Descriptions 3387
28-76. CRC_WDTOPLD2 Register Field Descriptions 3388
28-77. CRC_BCTOPLD2 Register Field Descriptions 3389
28-78. PSA_SIGREGL2 Register Field Descriptions 3390
28-79. PSA_SIGREGH2 Register Field Descriptions 3391
28-80. CRC_REGL2 Register Field Descriptions 3392
28-81. CRC_REGH2 Register Field Descriptions 3393
28-82. PSA_SECSIGREGL2 Register Field Descriptions 3394
28-83. PSA_SECSIGREGH2 Register Field Descriptions 3395
28-84. RAW_DATAREGL2 Register Field Descriptions 3396
28-85. RAW_DATAREGH2 Register Field Descriptions 3397
28-86. CRC_PCOUNT_REG3 Register Field Descriptions 3398
28-87. CRC_SCOUNT_REG3 Register Field Descriptions 3399
28-88. CRC_CURSEC_REG3 Register Field Descriptions 3400
28-89. CRC_WDTOPLD3 Register Field Descriptions 3401
28-90. CRC_BCTOPLD3 Register Field Descriptions 3402
28-91. PSA_SIGREGL3 Register Field Descriptions 3403
28-92. PSA_SIGREGH3 Register Field Descriptions 3404
28-93. CRC_REGL3 Register Field Descriptions 3405
28-94. CRC_REGH3 Register Field Descriptions 3406
28-95. PSA_SECSIGREGL3 Register Field Descriptions 3407
28-96. PSA_SECSIGREGH3 Register Field Descriptions 3408
28-97. RAW_DATAREGL3 Register Field Descriptions 3409
28-98. RAW_DATAREGH3 Register Field Descriptions 3410
28-99. CRC_PCOUNT_REG4 Register Field Descriptions 3411
28-100. CRC_SCOUNT_REG4 Register Field Descriptions 3412
28-101. CRC_CURSEC_REG4 Register Field Descriptions 3413
28-102. CRC_WDTOPLD4 Register Field Descriptions 3414
28-103. CRC_BCTOPLD4 Register Field Descriptions 3415
28-104. PSA_SIGREGL4 Register Field Descriptions 3416
28-105. PSA_SIGREGH4 Register Field Descriptions 3417
28-106. CRC_REGL4 Register Field Descriptions 3418
28-107. CRC_REGH4 Register Field Descriptions 3419
28-108. PSA_SECSIGREGL4 Register Field Descriptions 3420
28-109. PSA_SECSIGREGH4 Register Field Descriptions 3421
28-110. RAW_DATAREGL4 Register Field Descriptions 3422
28-111. RAW_DATAREGH4 Register Field Descriptions 3423
28-112. MCRC_BUS_SEL Register Field Descriptions 3424
28-113. Register Base Address (16xx/18xx) 3425
28-114. Register Base Address (14xx) 3425
28-115. PBIST Registers 3429
28-116. PBIST Access Type Codes 3429
28-117. PBIST_DLR Register Field Descriptions 3430
28-118. PBIST_PACT Register Field Descriptions 3431

28-119. PBIST_ID Register Field Descriptions 3432
28-120. PBIST_OVR Register Field Descriptions 3433
28-121. PBIST_FSFR0 Register Field Descriptions 3434
28-122. PBIST_FSFR1 Register Field Descriptions 3435
28-123. PBIST_FSRCR0 Register Field Descriptions 3436
28-124. PBIST_FSRCR1 Register Field Descriptions 3437
28-125. PBIST_ROM Register Field Descriptions 3438
28-126. PBIST_ALGO Register Field Descriptions 3439
28-127. PBIST_RINFOL Register Field Descriptions 3440
28-128. PBIST_RINFOU Register Field Descriptions 3441
28-129. Memory Map for 16xx/18xx 3441
28-130. Memory Map for 14xx 3442
28-131. ROM Organization for 2 Intervals 3447
28-132. STC Registers 3454
28-133. STC Access Type Codes 3455
28-134. STCGCR0 Register Field Descriptions 3456
28-135. STCGCR1 Register Field Descriptions 3457
28-136. STCTPR Register Field Descriptions 3458
28-137. STC_CADDR Register Field Descriptions 3459
28-138. STCCICR Register Field Descriptions 3460
28-139. STCGSTAT Register Field Descriptions 3461
28-140. STCFSTAT Register Field Descriptions 3462
28-141. STCSCSCR Register Field Descriptions 3463
28-142. STC_CADDR2 Register Field Descriptions 3464
28-143. STC_CLKDIV Register Field Descriptions 3465
28-144. STC_SEGPLR Register Field Descriptions 3466
28-145. SEG0_START_ADDR Register Field Descriptions 3467
28-146. SEG1_START_ADDR Register Field Descriptions 3468
28-147. SEG2_START_ADDR Register Field Descriptions 3469
28-148. SEG3_START_ADDR Register Field Descriptions 3470
28-149. CORE1_CURMISR_0 Register Field Descriptions 3471
28-150. CORE1_CURMISR_1 Register Field Descriptions 3472
28-151. CORE1_CURMISR_2 Register Field Descriptions 3473
28-152. CORE1_CURMISR_3 Register Field Descriptions 3474
28-153. CORE1_CURMISR_4 Register Field Descriptions 3475
28-154. CORE1_CURMISR_5 Register Field Descriptions 3476
28-155. CORE1_CURMISR_6 Register Field Descriptions 3477
28-156. CORE1_CURMISR_7 Register Field Descriptions 3478
28-157. CORE1_CURMISR_8 Register Field Descriptions 3479
28-158. CORE1_CURMISR_9 Register Field Descriptions 3480
28-159. CORE1_CURMISR_10 Register Field Descriptions 3481
28-160. CORE1_CURMISR_11 Register Field Descriptions 3482
28-161. CORE1_CURMISR_12 Register Field Descriptions 3483
28-162. CORE1_CURMISR_13 Register Field Descriptions 3484
28-163. CORE1_CURMISR_14 Register Field Descriptions 3485
28-164. CORE1_CURMISR_15 Register Field Descriptions 3486
28-165. CORE1_CURMISR_16 Register Field Descriptions 3487
28-166. CORE1_CURMISR_17 Register Field Descriptions 3488
28-167. CORE1_CURMISR_18 Register Field Descriptions 3489

28-168. CORE1_CURMISR_19 Register Field Descriptions 3490
28-169. CORE1_CURMISR_20 Register Field Descriptions 3491
28-170. CORE1_CURMISR_21 Register Field Descriptions 3492
28-171. CORE1_CURMISR_22 Register Field Descriptions 3493
28-172. CORE1_CURMISR_23 Register Field Descriptions 3494
28-173. CORE1_CURMISR_24 Register Field Descriptions 3495
28-174. CORE1_CURMISR_25 Register Field Descriptions 3496
28-175. CORE1_CURMISR_26 Register Field Descriptions 3497
28-176. CORE1_CURMISR_27 Register Field Descriptions 3498
28-177. CORE2_CURMISR_0 Register Field Descriptions 3499
28-178. CORE2_CURMISR_1 Register Field Descriptions 3500
28-179. CORE2_CURMISR_2 Register Field Descriptions 3501
28-180. CORE2_CURMISR_3 Register Field Descriptions 3502
28-181. CORE2_CURMISR_4 Register Field Descriptions 3503
28-182. CORE2_CURMISR_5 Register Field Descriptions 3504
28-183. CORE2_CURMISR_6 Register Field Descriptions 3505
28-184. CORE2_CURMISR_7 Register Field Descriptions 3506
28-185. CORE2_CURMISR_8 Register Field Descriptions 3507
28-186. CORE2_CURMISR_9 Register Field Descriptions 3508
28-187. CORE2_CURMISR_10 Register Field Descriptions 3509
28-188. CORE2_CURMISR_11 Register Field Descriptions 3510
28-189. CORE2_CURMISR_12 Register Field Descriptions 3511
28-190. CORE2_CURMISR_13 Register Field Descriptions 3512
28-191. CORE2_CURMISR_14 Register Field Descriptions 3513
28-192. CORE2_CURMISR_15 Register Field Descriptions 3514
28-193. CORE2_CURMISR_16 Register Field Descriptions 3515
28-194. CORE2_CURMISR_17 Register Field Descriptions 3516
28-195. CORE2_CURMISR_18 Register Field Descriptions 3517
28-196. CORE2_CURMISR_19 Register Field Descriptions 3518
28-197. CORE2_CURMISR_20 Register Field Descriptions 3519
28-198. CORE2_CURMISR_21 Register Field Descriptions 3520
28-199. CORE2_CURMISR_22 Register Field Descriptions 3521
28-200. CORE2_CURMISR_23 Register Field Descriptions 3522
28-201. CORE2_CURMISR_24 Register Field Descriptions 3523
28-202. CORE2_CURMISR_25 Register Field Descriptions 3524
28-203. CORE2_CURMISR_26 Register Field Descriptions 3525
28-204. CORE2_CURMISR_27 Register Field Descriptions 3526

Trademarks

ICEPick-D, TMS320C674x, TMS320C64x+, C64x+, TMS320C64x+, C64x+, TMS320C64x+, C64x+ are trademarks of Texas Instruments.
JTAG is a registered trademark of JTAG Technologies B.V..

14xx

Topic Page

1.1 14xx Introduction..... 135
1.2 14xx Memory Map..... 138
1.3 14xx Integration..... 143

1.1 14xx Introduction

1.1.1 14xx Overview

The 14xx is highly integrated single-chip RADAR device in TI's 45-nm low-power RFCMOS technology, a FCBGA 0.65-mm pitch automotive-grade package.

1.1.1.1 Features

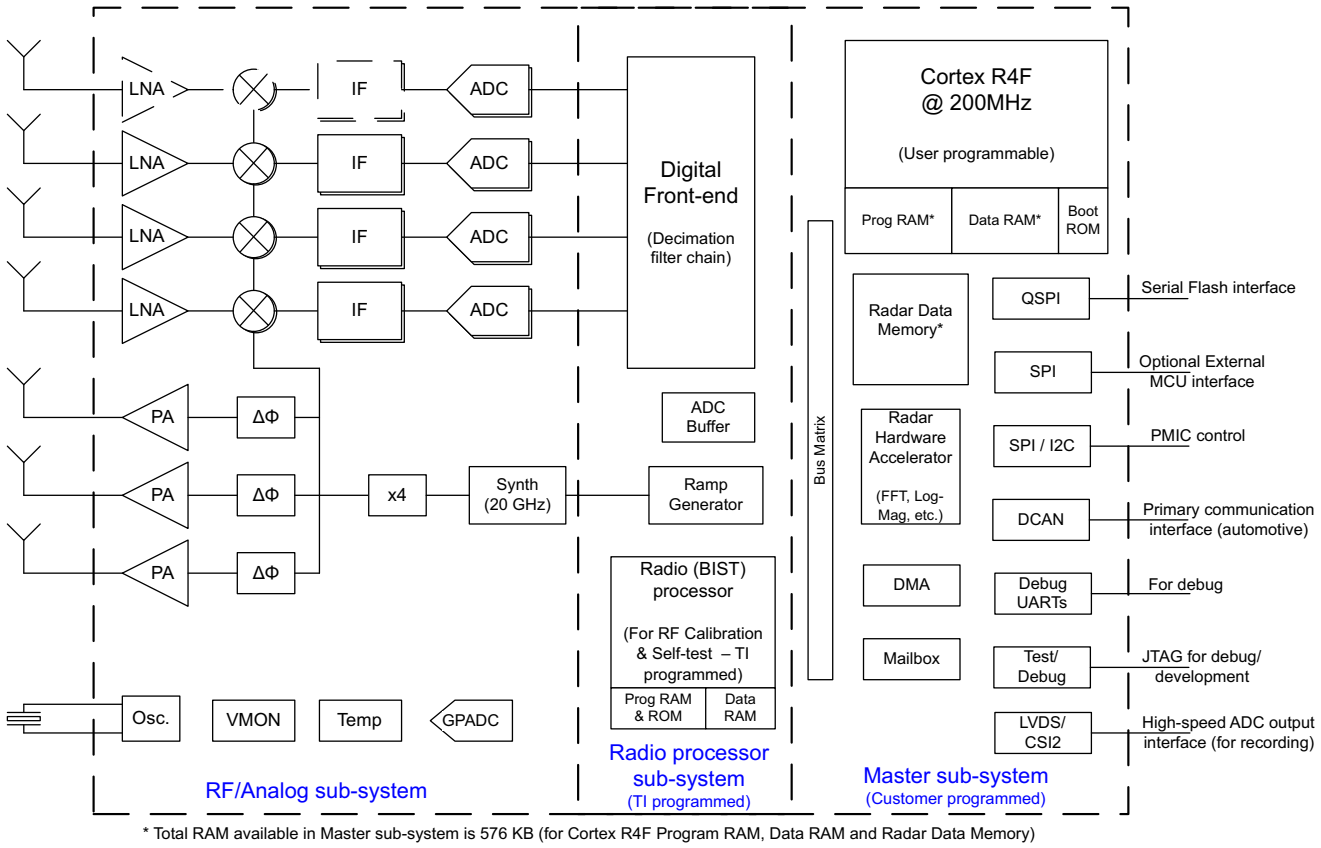
- Frequency-Modulated Continuous Wave Radio Frequency Transceiver With 76-81-GHz Band
- Supports Three Transmitter Chains and Four Receiver Chains
- Chirp Profiles With Programmable Period and Slope
- 40-MHz XTAL/OSC Reference Input Clock
- 12, 14, and 16-bit Real/Complex ADC With Variable Baseband ADC Sampling Rates up to 18.75 MHz at 12-bits Complex
- Cortex R4F at 200-MHz Application Processor for Control Functionality and Safety-Critical Algorithms
- Cortex R4F– Radio Processor at 200 MHz for Continuous Monitoring and Calibration of Analog/RF Functionality
- High-Performance Data Transfer With Multiple DMA and EDMA-TPCC Engines
- CAN Support for ECU Interface
- QSPI Serial Flash Support
- MIBSPI, SPI, I2C, and UART Serial Interfaces Support
- Four Data, One Clock Lane Serial LVDS Interface Support

1.1.2 14xx Description

1.1.2.1 Block Diagram

Figure 1-1 shows the block diagram of the 14xx device.

Figure 1-1. 14xx Block Diagram



* Total RAM available in Master sub-system is 576 KB (for Cortex R4F Program RAM, Data RAM and Radar Data Memory)

Table 1-1. 14xx Acronyms

ADC	Analog-to-Digital Convertor
CSI-2	Camera Serial Interface 2
DCAN	Controller Area Network
DMA	Direct Memory Access
GPADC	General Purpose Analog-to-Digital Convertor
I2C	Inter-Integrated Circuit
IF	Intermediate Frequency
LNA	Low-Noise Amplifier
LVDS	Low Voltage Differential Signaling
Osc	Oscillator
PA	Power Amplifier
QSPI	Quad Serial Peripheral Interface
SPI	Serial Peripheral Interface
UART	Univeral Asynchronous Receiver/Transmitter
VMON	Voltage Monitor
ΔΦ	Phase Modulator

1.1.2.2 Radar Subsystem

The RADAR subsystem is responsible for the RF and analog functionality of the device. The subsystem incorporates a built-in self-test processor for the continuous motoring and calibration of the analog and RF modules. The subsystem consists of:

- FMCW transceiver
  - Integrated PLL, transmitter, receiver, baseband, and A2D
  - 76-81-GHz coverage with 4-GHz available bandwidth
  - Four receive channels
  - Two transmit channels
  - Ultra-accurate chirp engine based on fractional-n PLL
  - 12,14, or 16-bit complex analog to digital converter
- Radio processor for built-in calibration and self-test
  - ARM Cortex R4F-based radio control system
  - Built-in firmware (ROM)
  - Self-calibrating system across frequency and temperature

This subsystem is TI-programmed with an API interface to the on-chip Cortex-R4F application processor.

1.1.2.3 Master Subsystem

The master subsystem consists of the following features:

- Dual Cortex-R4F core ARMv7-R, VFPv3-D16, and ARMv7 debug architecture
- Tightly-coupled memories
  - 96 KB of ROM
  - 128KB of program RAM with ECC
  - 64KB of data RAM with ECC
- Hardware auto-initialization of the memories
- Vectored interrupt manager for prioritizing and controlling the interrupts for different sources

1.1.2.3.1 Serial Interfaces

- One DCAN controller supporting bit rates of up to 1 Mbit/s, and compliant to the controller area network (CAN) 2.0B protocol specification
- One I2C controller module with rates up to 400 kbps
- Two high-speed synchronous serial input/output MIBSPI modules
- Two serial communication interface (SCI) modules implementing standard universal asynchronous receiver-transmitter (UART)
- One quad SPI module support with maximum rate of 40 MHz

1.1.2.3.2 System Peripherals

- Multiple general-purpose input/output (GPIO) modules
- Direct memory access modules for high-performance data transfers
- One watchdog timer and a general purpose timer implemented by the real-time interrupt (RTI) modules
- Mailbox module for interprocessor communication
- System reset and control module, which contains registers for the following functions:
  - Status
  - Efuse logic
  - I/O configuration
  - PAD configuration
  - System boot decoding logic

1.1.2.4 Functional Safety Deliverables

See the Device Safety Manual for supported features.

1.1.2.5 On-Chip Debug Support

The on-chip debug support has the following features:

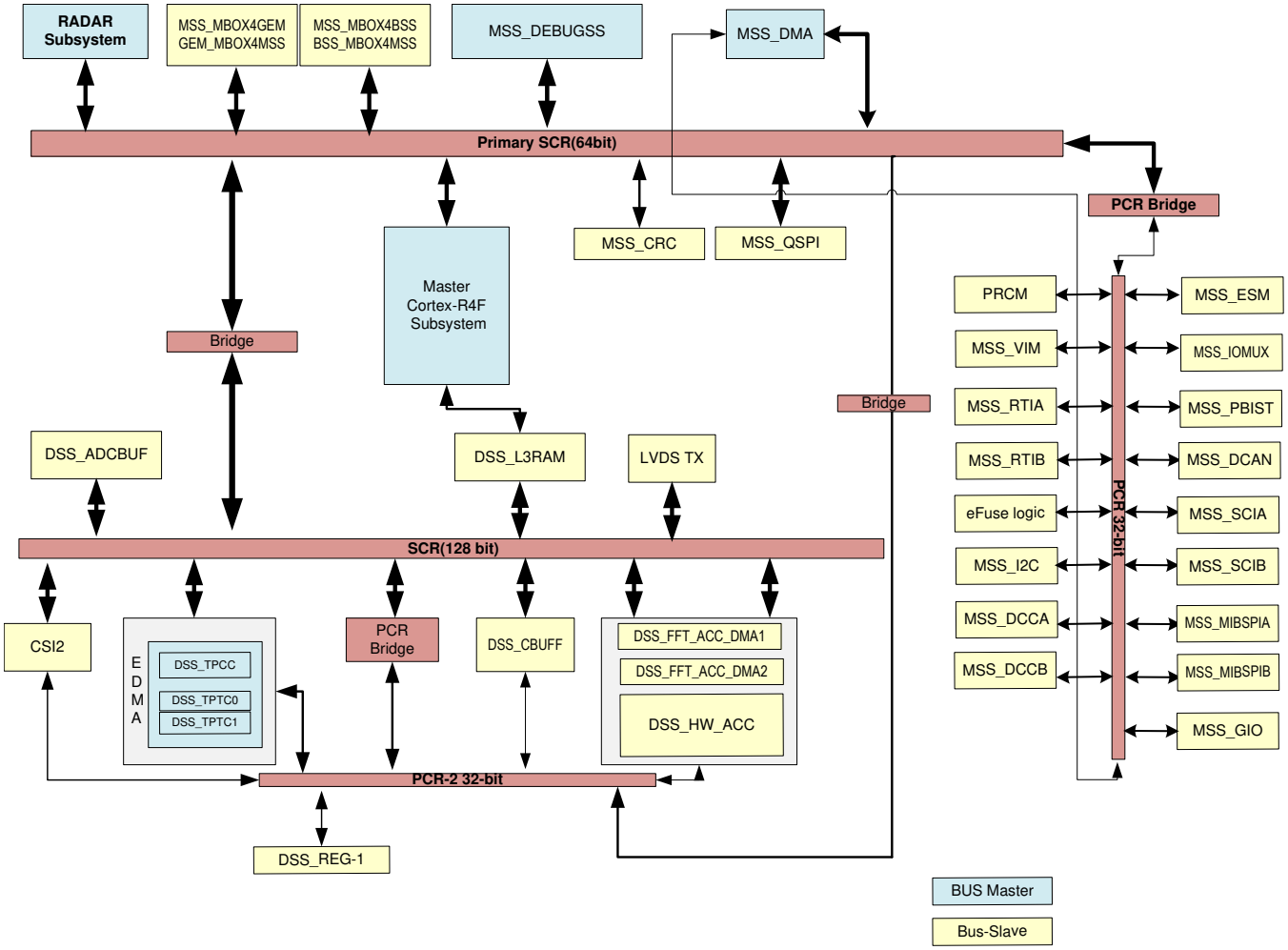
- The following device cores can be debugged through Code Composer Studio (CCS):
  - Cortex-R4F
- Target debugging using IEEE1149.1 (JTAG®) port
- The debug subsystem includes:
  - IEEE1149.7 adapter
  - Generic TAP for emulation and test control (ICEPick-D™)
  - Debug access port (DAP)
  - Embedded trace macro (ETM)
  - Trace port interface Unit (TPIU)
  - Embedded trace buffer (ETB)

1.2 14xx Memory Map

1.2.1 System Interconnect

The device implements a system interconnect based on TI's common bus architecture, comprising of VBUSM and VBUSP protocols. Figure 1-2 shows the interconnect diagram.

Figure 1-2. System Interconnect



The system interconnect is designed for the high-performance needs of the system. Its divided into interconnect systems local to each subsystem: the RADAR subsystem and master subsystem. The interconnection of all these subsystems is shown in .

In the master subsystem, the primary VBUSM SCR is responsible for managing the arbitration priority between accesses from multiple masters to each of the slaves. The arbitration priority is always round-robin.

The master subsystem has PCR interconnect that manages the accesses to the peripheral registers and peripheral memories, and provides a global reset for all peripherals. It also supports the capability to selectively enable or disable the clock for each peripheral individually. The PCR also manages the accesses to the system module registers required to configure the device clocks, interrupts, and so forth. The system module registers include status flags for indicating exception conditions – resets, aborts, errors, and interrupts.

1.2.2 Master Subsystem Cortex-R4F Memory Map

Table 3-2 shows the master subsystem, Cortex-R4F memory map.

Table 1-2. Master Subsystem, Cortex-R4F Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_TCMA_ROM	0x0000_0000	0x0001_7FFF	96KiB	MSS_TCMA_ROM (TCMA) Program ROM (refer to ROM Eclipsing section)
Reserved	0x0001 8000	0x001F_FFFF		Reserved (refer to ROM Eclipsing section)
MSS_TCMA_RAM	0x0020_0000	0x07FF_FFFF	128KiB	MSS_TCMA_RAM (TCMA) size varies based on device and DSS_L3 (L3) sharing options configured (refer to ROM Eclipsing section)
MSS_TCMB	0x0800_0000	0x0802_FFFF	64KiB	MSS_TCMB (TCMB) Data RAM
Reserved	0x0C20 0000	0x4FFF_FFFF		Reserved
DSS_TPTC0	0x5000_0000	0x5000_03FF	1 KiB	DSS_TPTC0 (EDMA TPTC0) module configuration space
DSS_REG	0x5000_0400	0x5000_07FF	584B	DSS_REG (DSPSS) control module registers
DSS_TPTC1	0x5000_0800	0x5000_0BFF	1 KiB	DSS_TPTC1 (EDMA TPTC1) module configuration space
Reserved	0x5000 0C00	0x5000_FFFF		Reserved
DSS_TPCC	0x5001_0000	0x5001_FFFF	64 KiB	DSS_TPCC (EDMA TPCC0) module configuration space
Reserved	0x5002 0000	0x5005_FFFF		Reserved
CSI2	0x5006_0000	0x5006_03FF	512B	CSI2 (CSI2) configuration register space (refer to HSI chapter)
CSI2 DHY	0x5006_0200	0x5006_05FF	64B	CSI2-DHY (CSI2) space (refer to HSI chapter)
Reserved	0x5006_0600	0x5006_FFFF		Reserved
DSS_CBUFF (CSI/LVDS CBUF)	0x5007_0000	0x5007_FFFF	564B	DSS_CBUFF (CBUFF) module configuration registers (refer to HSI chapter)
DSS_HW_ACC_PARAM	0x5008_0000	0x5008_07FF	512B	DSS_HW_ACC_PARAM (HWA) FFT accelerator PARAM memory
DSS_HW_ACC_STATIC	0x5008_0800	0x5008_0FFF	264B	DSS_HW_ACC_STATIC (HWA) FFT accelerator configuration registers
DSS_HW_ACC_WIN	0x5008_1000	0x5008_FFFF	4KiB	DSS_HW_ACC_WIN (HWA) FFT accelerator Window registers
Reserved	0x5009_0000	0x50FF_FFFF		Reserved
DSS_L3RAM	0x5100_0000	0x51FF_FFFF	384KiB	DSS_L3RAM (L3) shared memory space
DSS_ADCBUF	0x5200_0000	0x5201_FFFF	16KiB	DSS_ADCBUF (ADC) buffer memory space
DSS_CBUFF_FIFO	0x5202_0000	0x5202_7FFF	16KiB	DSS_CBUFF_FIFO (CBUFF) Common buffer FIFO space (refer to HSI chapter)
Reserved	0x5202 8000	0x5202_FFFF		Reserved
DSS_FFT_ACC_DMA1	0x5203_0000	0x5203_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator Memory -1 space
DSS_FFT_ACC_DMA2	0x5203_8000	0x5206_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator Memory -2 space
DSS_REG_VBUSM	0x5207_0000	0x5207_07FF	128B	DSS_REG_VBUSM 128-bit SCR configuration port
DSS_DATA_TXFR_RAM	0x5207_0800	0xBFFF_FFFF	8KiB	DSS_DATA_TXFR_RAM memory space
EXT_FLASH	0xC000_0000	0xC07F_FFFF	8MB	MSS_QSPI (QSPI) flash memory space
MSS_QSPI	0xC080_0000	0xC0FF_FFFF	116B	MSS_QSPI (QSPI) module configuration registers
MSS_MBOX4BSS	0xF060_1000	0xF060_17FF	2KiB	MSS_MBOX4BSS mailbox memory space
BSS_MBOX4MSS	0xF060_2000	0xF060_27FF	2KiB	BSS_MBOX4MSS mailbox memory space
Reserved	0xF060_4000	0xF060_7FFF		Reserved

Table 1-2. Master Subsystem, Cortex-R4F Memory Map (continued)

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
BSS_MBOX4MSS_REG	0xF060_8000	0xF060_80FF	188B	BSS_MBOX4MSS_REG mailbox Configuration registers
Reserved	0xF060_8100	0xF060_85FF		Reserved
MSS_MBOX4BSS_REG	0xF060_8600	0xF060_86FF	188B	MSS_MBOX4BSS_REG mailbox Configuration registers
Reserved	0xFCF7_8C00	0xFCFF_0FFF		Reserved
MSS_PCR2	0xFCFF_1000	0xFCFF_17FF	1KiB	MSS_PCR2 (PCR_2) interconnect configuration port
Reserved	0xFCFF_F600	0xFDFF_FFFF		Reserved
MSS_MCRC	0xFE00_0000	0xFEFF_FFFF	16KiB	MSS_MCRC (CRC) module configuration registers (refer to Safety chapter)
Reserved	0xFF0C_0000	0xFF0D_FFFF		Reserved
MSS_MIBSPIA_TXRAM	0xFF0E_0000	0xFF0E_01FF	0.5KB	MSS_MIBSPIA_TXRAM (MIBSPIA) memory space
MSS_MIBSPIA_RXRAM	0xFF0E_0200	0xFF0E_03FF	0.5KB	MSS_MIBSPIA_RXRAM (MIBSPIA) memory space
MSS_DCAN_MEM	0xFF1E_0000	0xFF1F_FFFF	128KB	MSS_DCAN_MEM (CAN) RAM memory space
Reserved	0xFF50_0000	0xFF9F_FFFF		Reserved
MSS_DEBUGSS	0xFFA0_0000	0xFFAF_FFFF	244KiB	MSS_DEBUGSS (Debug subsystem) memory space and registers
MSS_PCR	0xFFF7_8000	0xFFF7_87FF	1KiB	MSS_PCR (PCR_1) interconnect configuration port
Reserved	0xFFF7 A000	0xFFF7_BBFF		Reserved
MSS_GIO	0xFFF7_BC00	0xFFF7_BDFF	180B	MSS_GIO (GIO) module configuration registers
Reserved	0xFFF7 C800	0xFFF7_D3FF		Reserved
MSS_I2C	0xFFF7_D400	0xFFF7_D4FF	112B	MSS_I2C (I2C) module configuration registers
MSS_DCAN	0xFFF7_DC00	0xFFF7_DDFF	512B	MSS_DCAN (CAN) module configuration registers
MSS_SCIA (UART)	0xFFF7_E500	0xFFF7_E5FF	148B	MSS_SCIA (SCIA/UART) module configuration registers
MSS_SCIB (UART)	0xFFF7_E700	0xFFF7_E7FF	148B	MSS_SCIB (SCIB/UART) module configuration registers
MSS_MIBSPIA	0xFFF7_F400	0xFFF7_F5FF	512B	MSS_MIBSPIA (MIBSPIA) module configuration registers
MSS_SPIB	0xFFF7_F600	0xFFF7_F7FF	512B	MSS_SPIB (SPIB) module configuration registers
MSS_DMA_RAM	0xFFF8_0000	0xFFF8_0FFF	4KB	MSS_DMA_RAM (DMA1) RAM memory space
MSS_VIM_MEM	0xFFF8_2000	0xFFF8_2FFF	2KB	MSS_VIM_MEM (VIM) RAM memory space
Reserved	0xFFF8 C000	0xFFFF_E0FF		Reserved
MSS_TOPRCM	0xFFFF_E100	0xFFFF_E2FF	756B	MSS_TOPRCM TOP Level Reset, Clock management registers
MSS_PBIST	0xFFFF_E400	0xFFFF_E5FF	464B	MSS_PBIST (PBIST) module configuration registers (refer to Safety chapter)
MSS_STC	0xFFFF_E600	0xFFFF_E7FF	284B	MSS_STC (STC) module configuration registers (refer to Safety chapter)
MSS_IOMUX	0xFFFF_EA00	0xFFFF_EBFF	512KiB	MSS_IOMUX (IOMUX) module registers
MSS_DCCA	0xFFFF_EC00	0xFFFF_ECFF	44B	MSS_DCCA (DCCA) module configuration registers (refer to Safety chapter)

Table 1-2. Master Subsystem, Cortex-R4F Memory Map (continued)

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_RTIB (WDT/RTIB)	0xFFFF_EE00	0xFFFF_EEFF	192B	MSS_RTIB (WDT/RTIB) module register space
MSS_DMA_REG	0xFFFF_F000	0xFFFF_F3FF	1KiB	MSS_DMA_REG (DMA1) module configuration registers
MSS_DCCB	0xFFFF_F400	0xFFFF_F4FF	44B	MSS_DCCB (DCCB) module configuration registers (refer to Safety chapter)
MSS_ESM	0xFFFF_F500	0xFFFF_F5FF	156B	MSS_ESM (ESM) module configuration registers (refer to Safety chapter)
Reserved	0xFFFF_F600	0xFFFF_F7FF		Reserved
MSS_GPCFG_REG	0xFFFF_F800	0xFFFF_FBFF	352B	MSS_GPCFG_REG (GPCFG) General purpose control registers
MSS_RTIA	0xFFFF_FC00	0xFFFF_FCFF	192B	MSS_RTIA (RTIA) module
MSS_VIM	0xFFFF_FD00	0xFFFF_FEFF	512B	MSS_VIM (VIM) module configuration registers
MSS_RCM	0xFFFF_FF00	0xFFFF_FFFF	256B	MSS_RCM (RCM) Reset, Clock management registers

1.2.2.1 Radar Subsystem Interface

The RADAR subsystem is accessible through a set of TI-implemented high-level API calls by the application running on the master CR4F.

1.2.3 EDMA Memory Map

Table 1-3. EDMA-TPTC Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_L3RAM	0x2000_0000	0x201F_FFFF	2MB	DSS_L3RAM (L3) shared memory space
DSS_ADCBUF	0x2100_0000	0x2100_7FFC	32KiB	DSS_ADCBUF (ADC) memory space
DSS_CBUFF_FIFO	0x2102_0000	0x2102_3FFC	16KiB	DSS_CBUFF_FIFO (CBUFF) memory space
DSS_FFT_ACC_DMA1	0x2103 0000	0x2103_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator memory-1 space
DSS_FFT_ACC_DMA2	0x2103 8000	0x401F_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator memory-2 space
MSS_TCMA_RAM	0x4020_0000	0x4023_FFFF	256KiB	MSS_TCMA_RAM (TCMA) Data RAM
MSS_TCMB	0x4800_0000	0x4802_FFFF	192KiB	MSS_TCMB (TCMB) Data RAM
MSS_SW_BUFFER	0x4C20_0000	0x4C20_1FFF	8KiB	MSS_SW_BUFFER (SWBUFF) Scratchpad memory
GEM_MBOX4MSS	0x5060_4000	0x5060_4000	2KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0x5060_5000	0x5060_5000	2KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0x5060_6000	0x5060_6000	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_7000	2KiB	BSS_MBOX4GEM mailbox memory space

1.3 14xx Integration

1.3.1 Cortex-R4F Subsystem

1.3.1.1 Tightly Coupled Memories

The total memory (RAM) available in the master subsystem is 576 KB. This is partitioned between the R4F program RAM, R4F data RAM, and radar data memory. The maximum usable size for R4F is 448 KB, and this is partitioned between the R4F's tightly coupled interfaces MSS_TCMA_RAM (320 KB) and MSS_TCMB (128 KB). Although the complete 448 KB is unified memory and can be used for program or data, typical applications use MSS_TCMA_RAM as program and MSS_TCMB as data memory.

The remaining memory, starting at a minimum of 128 KB, is available as radar data memory for storing the 'radar data cube'. The user can increase the radar data memory size in 64 KB increments, at the cost of corresponding reduction in R4F program or data RAM size. The maximum size of radar data memory possible is 384 KB. A few example configurations supported are listed in Table 1-4.

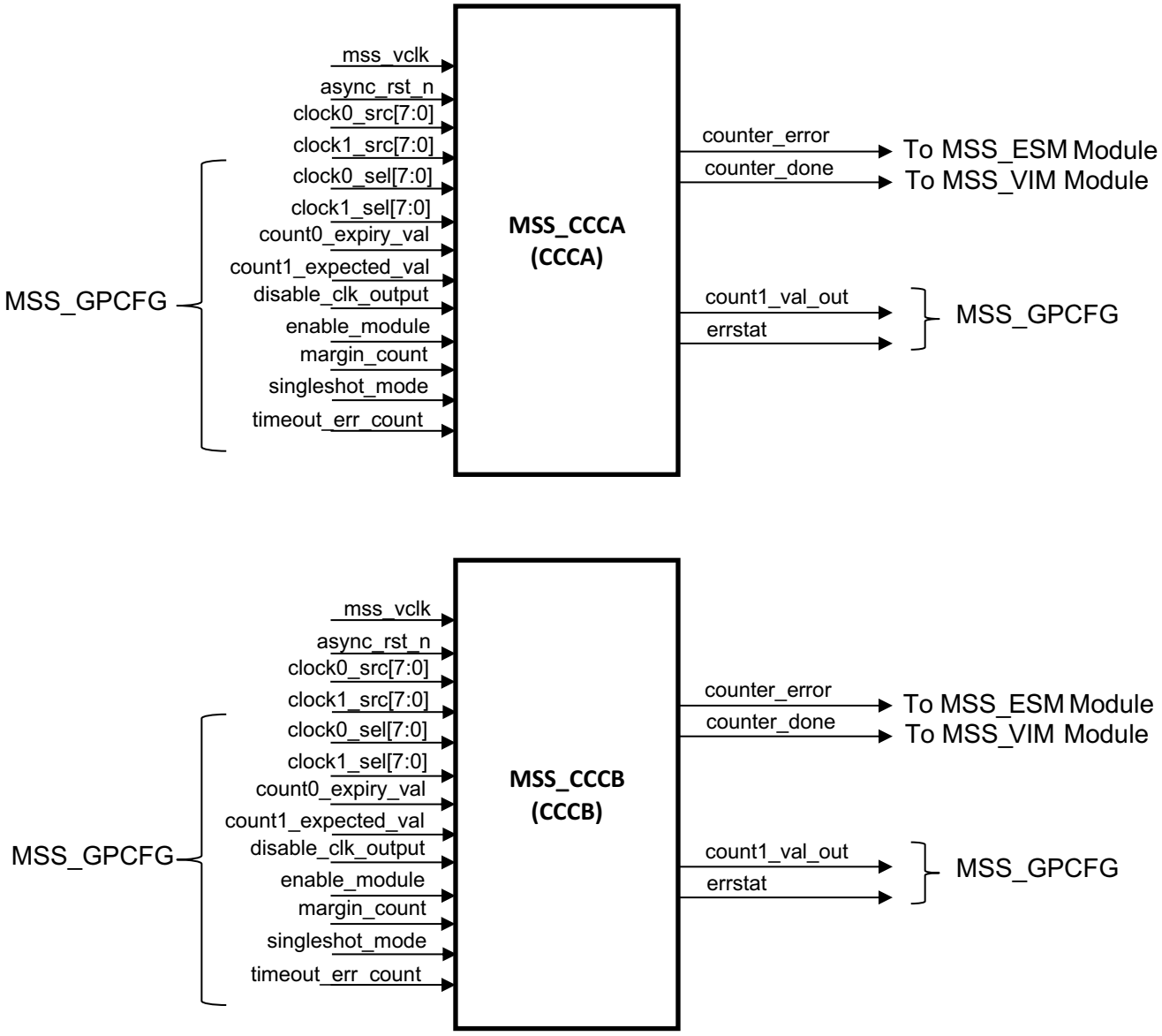
Table 1-4. Example Configurations

Option	R4F Program RAM (KB)	R4F Data RAM (KB)	Radar Data Memory (KB)
1	320	128	128
2	256	128	192
3	256	64	256
4	128	64	384

1.3.2 Clock Comparator

1.3.2.1 Core Clock Comparator (MSS_CCCA/MSS_CCCB)

Figure 1-3. Integration of MSS_CCCA and MSS_CCCB Modules



1.3.2.1.1 MSS_CCCA and MSS_CCCB Integration Connections

This device has two instances of CCC; MSS_CCCA (CCCA) and MSS_CCCB (CCCB). The clock connectivity information for these two instances are provided in Table 1-5. Configuration and status of this module is available through the MSS_GPCFG registers of the device.

Table 1-5. MSS_CCCA and MSS_CCCB Integration Connections

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
counter_error	ESM_GRP1[1]	ESM_GRP1[4]
counter_done	IRQ[80]	IRQ[81]

Table 1-5. MSS_CCCA and MSS_CCCB Integration Connections (continued)

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
clock0_src[0]	REFCLK	CR4_VCLK
clock0_src[1]	CPUCLK	Reserved
clock0_src[2]	RCCLK	BSSCLK
clock0_src[3]	RCCLK	QSPICLK
clock0_src[4]	RCCLK	Reserved
clock0_src[5]	RCCLK	REFCLK
clock0_src[6]	RCCLK	CPUCLK
clock0_src[7]	RCCLK	RCCLK
clock1_src[0]	REFCLK	PLLCLK_600
clock1_src[1]	Reserved	MSS_VCLK
clock1_src[2]	PLL 240Mhz	CPUCLK
clock1_src[3]	RCCLK	CR4_VCLK
clock1_src[4]	RCCLK	MSS_VCLK
clock1_src[5]	RCCLK	Reserved
clock1_src[6]	RCCLK	BSSCLK
clock1_src[7]	RCCLK	QSPICLK

1.3.2.2 Dual Clock Comparator (MSS_DCCA/MSS_DCCB)

Figure 1-4. MSS_DCCA/MSS_DCCB Integration Diagram

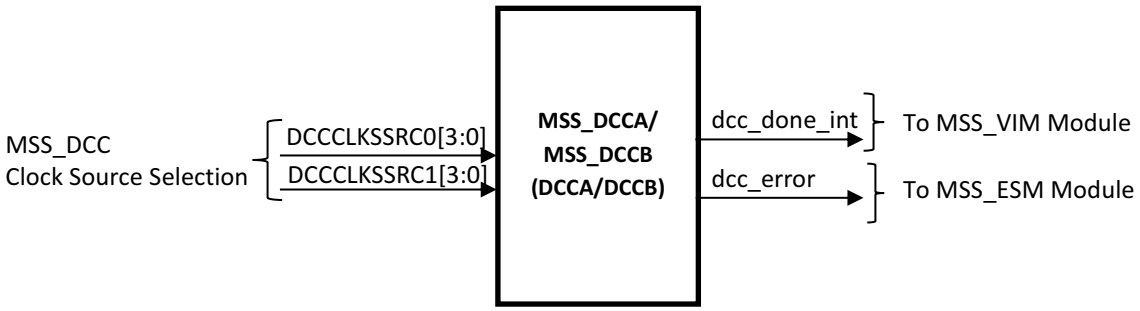


Table 1-6. MSS_DCCA Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - REF_CLK	0x0 - REF_CLK
0xA - PLL_600	0x1 - CPU_CLK
0x5 - PLL_240	0x2 to 0x7 - RC_CLK

Table 1-7. MSS_DCCB Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - PLL_600	0x0 - VCLK
0xA - VCLK	0x1 - DSS_CLK
0x5 - CPU_CLK	0x2 - BSS_CLK
	0x3 - QSPI_CLK
	0x4 - Reserved
	0x5 - REF_CLK
	0x6 - CPU_CLK
	0x7 - RC_CLK

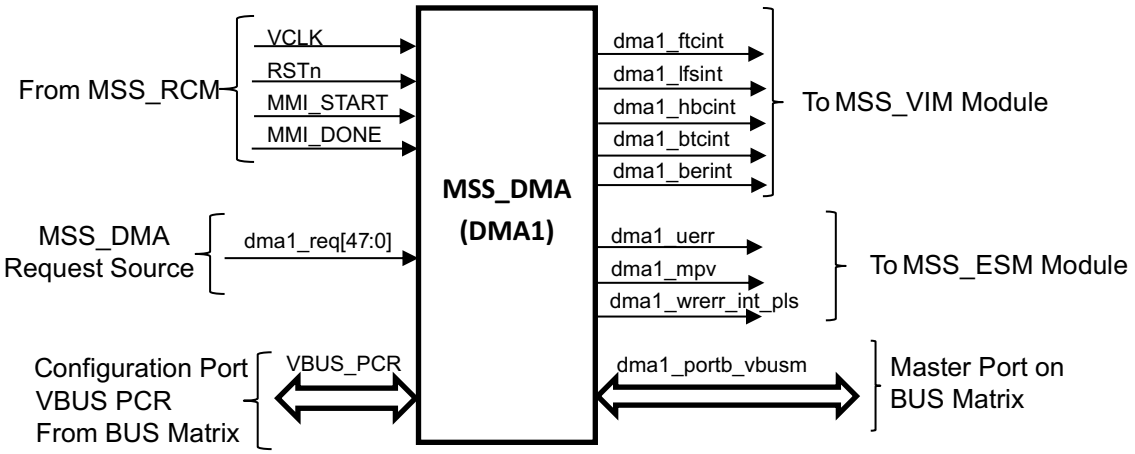
NOTE: Any values not mentioned are not used.

1.3.3 Direct Memory Access Controller (MSS_DMA)

1.3.3.1 MSS_DMA Integration Diagram

The device has one instance of DMA module, MSS_DMA. Integration of the MSS_DMA blocks in the device are shown in Figure 1-5.

Figure 1-5. Integration of MSS_DMA Module



1.3.3.2 MSS_DMA Features

- 64-bit OCP protocol to perform bus master accesses
- INCR-4 64-bit burst accesses
- Multithreading architecture allowing data of two different channel transfers to be interleaved during non-burst accesses
- 2-port configuration for parallel bus master
- Channels can be assigned to either high-priority queue or low-priority queue. Within each queue, fixed or round-robin priorities can be serviced
- Built-in ECC generation and evaluation logic for internal RAM-storing channel transfer information
- Supports multiple interrupt outputs for mapping to multiple interrupt controllers in multicore systems
- 48 requests can be mapped to any 32 channels
- Supports LE endianness
- External ECC Gen/Eval block of MSS_DMA support ECC generation for data transactions, and parity for address, and control signals (following Cortex-R5F standard)
- 8 MPU regions
- Channel-chaining capability
- Hardware and software MSS_DMA requests
- 8-, 16-, 32-, or 64-bit transactions supported
- Multiple addressing modes for source and destination (fixed, increment, offset)
- Auto-initiation

1.3.3.3 MSS_DMA Request Map

The MSS_DMA module has 32 channels and up to 42 hardware MSS_DMA requests. The module contains DREQASIx registers to map the MSS_DMA requests to the MSS_DMA channels. By default, channel 0 is mapped to request 0, channel 1 to request 1, and so forth.

Table 1-8. MSS_DMA Request Map

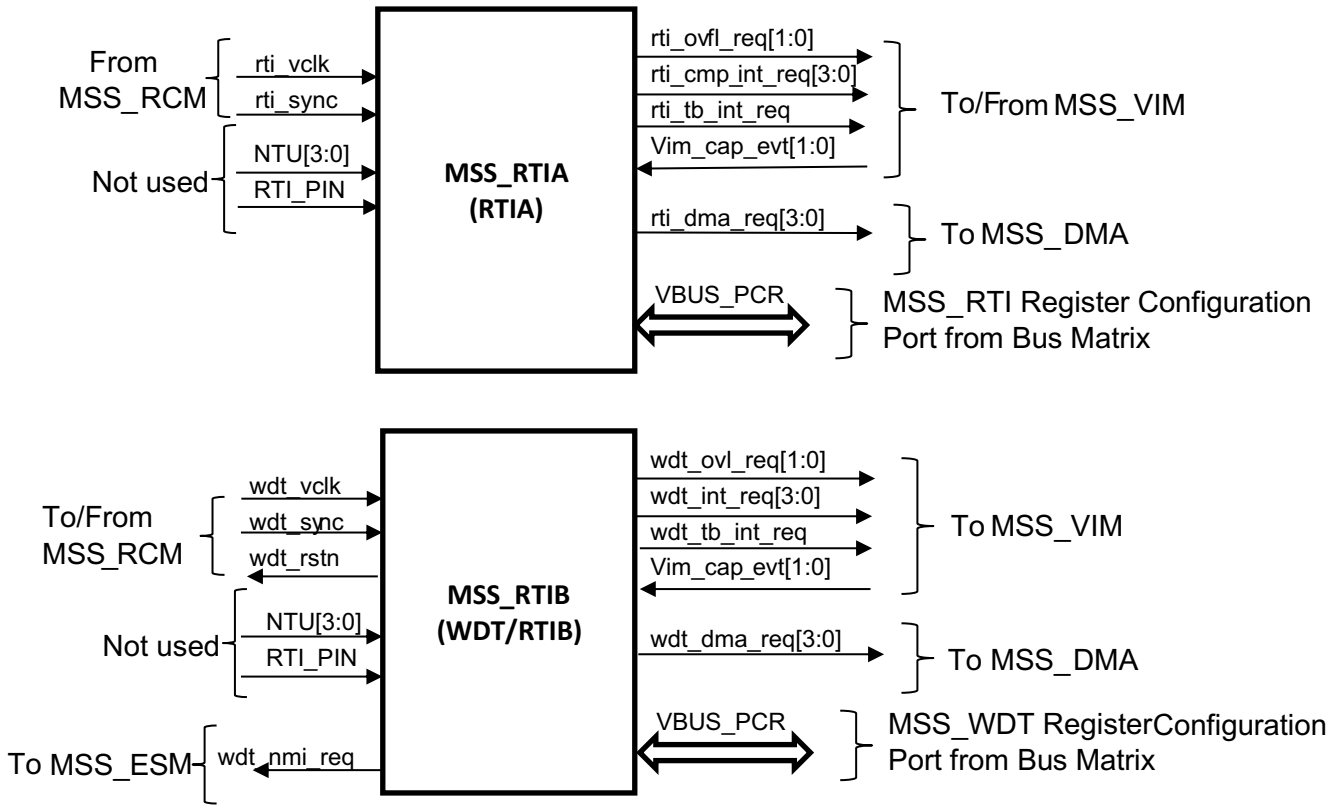
Module	DMA Request Sources	DMA Request
MSS_MIBSPIA	MSS_MIBSPIA Channel-1	DMAREQ[0]
MSS_MIBSPIA	MSS_MIBSPIA Channel-0	DMAREQ[1]
MSS_SPIB	MSS_SPIB Receive	DMAREQ[2]
MSS_SPIB	MSS_SPIB Transmit	DMAREQ[3]
MSS_QSPI	MSS_QSPI DMA request	DMAREQ[4]
MSS_MIBSPIA	MSS_MIBSPIA Channel-3	DMAREQ[5]
MSS_DCAN	MSS_DCAN IF2	DMAREQ[6]
DSS_CBUFF	DSS_CBUFF	DMAREQ[7]
MSS_DCAN	MSS_DCAN IF1	DMAREQ[8]
MSS_MIBSPIA	MSS_MIBSPIA Channel-5	DMAREQ[9]
MSS_I2C	MSS_I2C receive	DMAREQ[10]
MSS_I2C	MSS_I2C Transmit	DMAREQ[11]
MSS_RTIA	MSS_RTIA DMAREQ0	DMAREQ[12]
MSS_RTIA	MSS_RTIA DMAREQ1	DMAREQ[13]
Reserved	Reserved	DMAREQ[14]
Reserved	Reserved	DMAREQ[15]
MSS_DCAN	MSS_DCAN IF3	DMAREQ[16]
MSS_MIBSPIA	MSS_MIBSPIA Channel-9	DMAREQ[17]
MSS_RTIA	MSS_RTIA DMAREQ2	DMAREQ[18]
MSS_RTIA	MSS_RTIA DMAREQ3	DMAREQ[19]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ0	DMAREQ[20]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ1	DMAREQ[21]
MSS_MIBSPIA	MSS_MIBSPIA Channel-10	DMAREQ[22]
MSS_MIBSPIA	MSS_MIBSPIA Channel-11	DMAREQ[23]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ2	DMAREQ[24]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ3	DMAREQ[25]
MSS_MCRC	MSS_MCRC DMAREQ0	DMAREQ[26]
MSS_MCRC	MSS_MCRC DMAREQ1	DMAREQ[27]
MSS_SCIB (UART2)	MSS_SCIB (UART2) receive	DMAREQ[28]
MSS_SCIB (UART2)	MSS_SCIB (UART2) Transmit	DMAREQ[29]
MSS_SCI1 (UART1)	MSS_SCI1 (UART1) receive	DMAREQ[30]
MSS_SCI1 (UART1)	MSS_SCI1 (UART1) Transmit	DMAREQ[31]
MSS_GIO	MSS_GIO-0	DMAREQ[32]
MSS_GIO	MSS_GIO-1	DMAREQ[33]
MSS_GIO	MSS_GIO-2	DMAREQ[34]
Reserved	Reserved	DMAREQ[35]
MSS_MIBSPIA	MSS_MIBSPIA Channel-6	DMAREQ[36]
MSS_MIBSPIA	MSS_MIBSPIA Channel-7	DMAREQ[37]
MSS_MIBSPIA	MSS_MIBSPIA Channel-8	DMAREQ[38]
Reserved	Reserved	DMAREQ[39]
Reserved	Reserved	DMAREQ[40]
Reserved	Reserved	DMAREQ[41]
MSS_MIBSPIA	MSS_MIBSPIA Channel-12	DMAREQ[42]

Table 1-8. MSS_DMA Request Map (continued)

Module	DMA Request Sources	DMA Request
MSS_MIBSPIA	MSS_MIBSPIA Channel-13	DMAREQ[43]
MSS_MIBSPIA	MSS_MIBSPIA Channel-14	DMAREQ[44]
MSS_MIBSPIA	MSS_MIBSPIA Channel-15	DMAREQ[45]
MSS_GIO	MSS_GIO-14	DMAREQ[46]
MSS_GIO	MSS_GIO-15	DMAREQ[47]

1.3.4 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB)

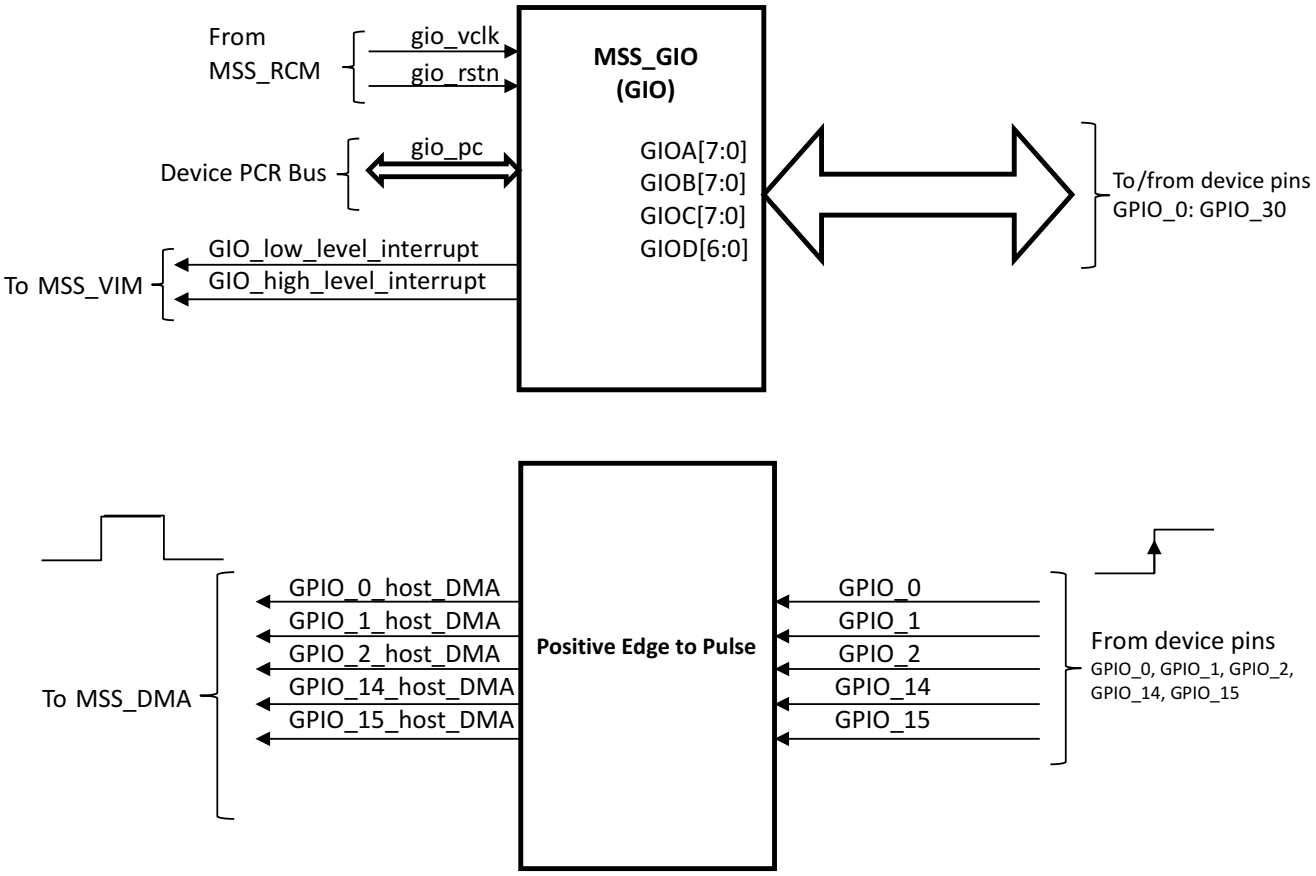
Figure 1-6. Integration of MSS_RTIA and MSS_RTIB, Using the RTI Module



1.3.5 General Purpose I/0 (MSS_GIO)

NOTE: Emulation mode and power-down mode (low-power mode) are not supported in the 14xx device.

Figure 1-7. Integration Block Diagram for MSS_GIO



1.3.6 Vectored Interrupt Manager (MSS_VIM)

1.3.6.1 Interrupt Request Assignments

Table 1-9. Interrupt Request Assignments

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
MSS_ESM	MSS_ESM high-level interrupt(NMI)	0
Reserved	Reserved	1
MSS_RTIA	MSS_RTIA compare interrupt 0	2
MSS_RTIA	MSS_RTIA compare interrupt 1	3
MSS_RTIA	MSS_RTIA compare interrupt 2	4
MSS_RTIA	MSS_RTIA compare interrupt 3	5
MSS_RTIA	MSS_RTIA overflow interrupt 0	6
MSS_RTIA	MSS_RTIA overflow interrupt 1	7
MSS_RTIA	MSS_RTIA time-base	8
MSS_GIO	MSS_GIO high-level interrupt	9
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 0	10
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt1	11
MSS_MIBSPIA	MSS_MIBSPIA level 0 Interrupt	12
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) Interrupt 2	13
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) Interrupt 3	14

Table 1-9. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) overflow interrupt 0	15
MSS_DCAN	MSS_DCAN level 0 interrupt	16
MSS_SPIB	MSS_SPIB level 0 Interrupt	17
Reserved	Reserved	18
MSS_MCRC	MSS_MCRC interrupt	19
MSS_ESM	MSS_ESM low-level interrupt	20
SYSTEM	Software-triggered interrupt 4	21
MSS Cortex R4F	MSS Cortex R4F interrupt PMU	22
MSS_GIO	MSS_GIO low-level interrupt	23
MSS_RTIB (WDT/RTIB)	WDT overflow interrupt 1	24
MSS_RTIB (WDT/RTIB)	WDT TB base interrupt	25
MSS_MIBSPIA	MSS_MIBSPIA level 0 interrupt	26
MSS_QSPI	MSS_QSPI interrupt	27
Reserved	Reserved	28
MSS_DCAN	MSS_DCAN level 1 interrupt	29
MSS_SPIB	MSS_SPIB level 1 interrupt	30
Reserved	Reserved	31
Reserved	Reserved	32
MSS_DMA	MSS_DMA frame transfer complete interrupt	33
MSS_DMA	MSS_DMA last frame transfer start interrupt	34
Reserved	Reserved	35
Reserved	Reserved	36
Reserved	Reserved	37
Reserved	Reserved	38
MSS_DMA	MSS_DMA half-block transfer complete interrupt	39
MSS_DMA	MSS_DMA block transfer complete interrupt	40
Reserved	Reserved	41
Reserved	Reserved	42
Reserved	Reserved	43
Reserved	Reserved	44
Reserved	Reserved	45
Reserved	Reserved	46
Reserved	Reserved	47
Reserved	Reserved	48
Reserved	Reserved	49
Reserved	Reserved	50
Reserved	Reserved	51
Reserved	Reserved	52
Reserved	Reserved	53
Reserved	Reserved	54
Reserved	Reserved	55
Reserved	Reserved	56
Reserved	Reserved	57
Reserved	Reserved	58

Table 1-9. Interrupt Request Assignments (continued)

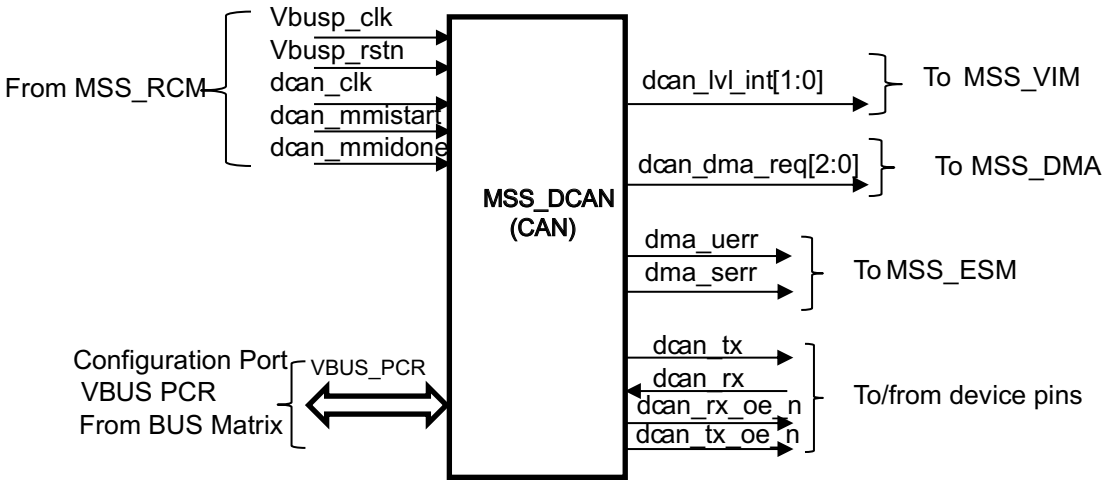
Module	VIM Interrupt Sources	Default VIM Interrupt Channel
Reserved	Reserved	59
Reserved	Reserved	60
Reserved	Reserved	61
Reserved	Reserved	62
Reserved	Reserved	63
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 0 interrupt	64
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 0 interrupt	65
MSS_I2C	MSS_I2C interrupt	66
Reserved	Reserved	67
Reserved	Reserved	68
Reserved	Reserved	69
MSS_DMA	MSS_DMA bus error interrupt	70
Reserved	Reserved	71
Reserved	Reserved	72
Reserved	Reserved	73
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 1 interrupt	74
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 1 interrupt	75
SYSTEM	Software-triggered interrupt 0	76
SYSTEM	Software-triggered interrupt 1	77
SYSTEM	Software-triggered interrupt 2	78
SYSTEM	Software-triggered interrupt 3	79
Reserved	Reserved	80
Reserved	Reserved	81
MSS_DCCA	MSS_DCCA (dual clock compare) module 1-done interrupt	82
MSS_DCCB	MSS_DCCB (dual clock compare) module 2-done interrupt	83
SYSTEM	Software-triggered interrupt 5	84
MSS_PBIST	MSS_PBIST done interrupt	85
Reserved	Reserved	86
Reserved	Reserved	87
Reserved	Reserved	88
Reserved	Reserved	89
Reserved	Reserved	90
Reserved	Reserved	91
Reserved	Reserved	92
Reserved	Reserved	93
Reserved	Reserved	94
MAILBOX	RADARSS to MSS mailbox interrupt	95
MAILBOX	RADARSS mailbox read complete interrupt sent from RADARASS to MSS	96
RADARSS	ADC valid fall interrupt	97
RADARSS	Frame start interrupt	98
RADARSS	Chirp start interrupt	99
RADARSS	Chirp end interrupt	100
RADARSS	Frame end interrupt	101
Reserved	Reserved	102
Reserved	Reserved	103

Table 1-9. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
Reserved	Reserved	104
BSS_CR4_STC	BSS_CR4_STC interrupt	105
Reserved	Reserved	106
Reserved	Reserved	107
Reserved	Reserved	108
Reserved	Reserved	109
Reserved	Reserved	110
Reserved	Reserved	111
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) interrupt	112
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) error interrupt	113
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) interrupt	114
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) error interrupt	115
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) interrupt	116
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) error interrupt	117
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) interrupt	118
DSS_ADCBUF	Falling edge of ADC valid interrupt	119
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) error interrupt	120
Reserved	Reserved	121
Reserved	Reserved	122
DSS_ADCBUF	Chirp available interrupt	123
Reserved	Reserved	124
DSS_HW_ACC	DSS_HW_ACC FFT accelerator -param done interrupt	125
DSS_HW_ACC	DSS_HW_ACC FFT accelerator - done interrupt	126
DSS_HW_ACC	DSS_HW_ACC FFT accelerator –access error interrupt	127

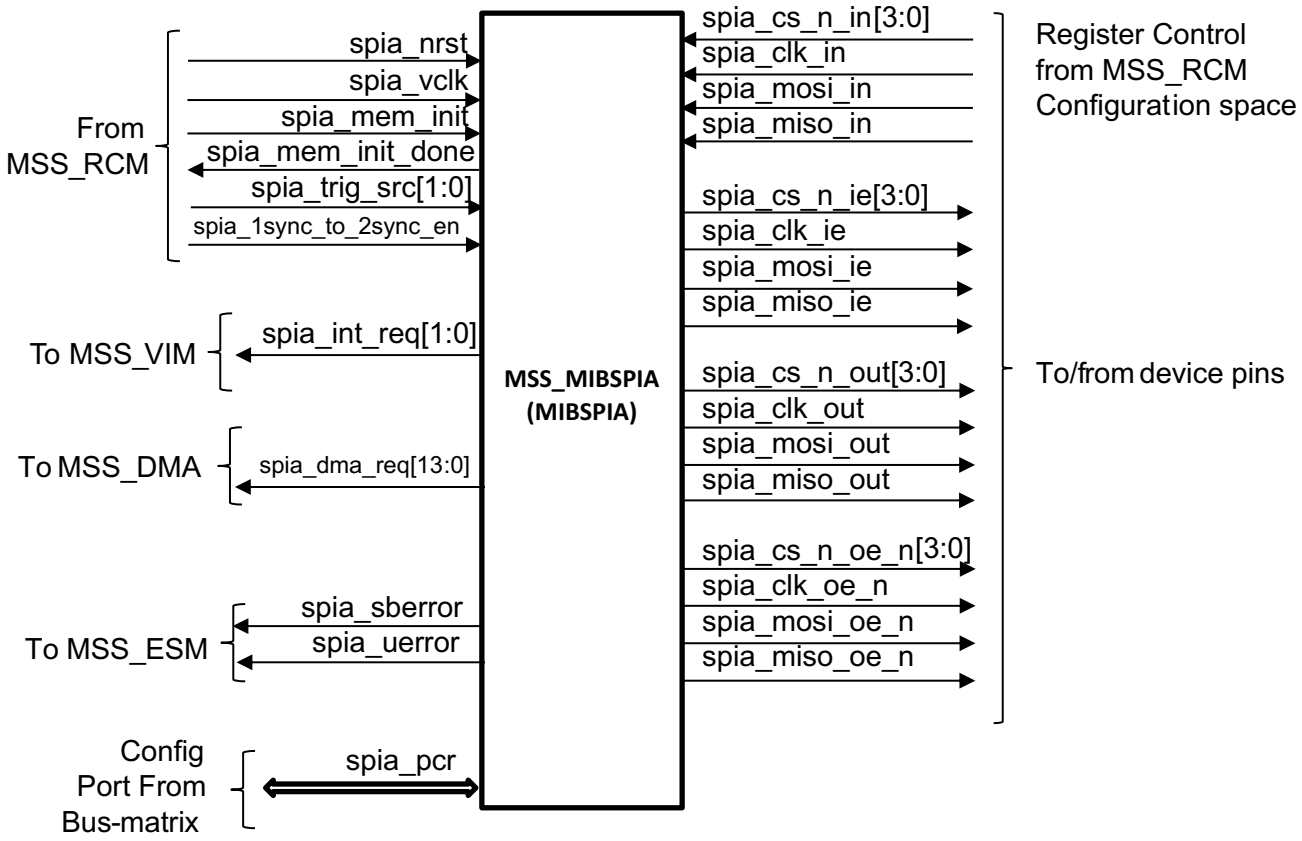
1.3.7 Controller Area Network (MSS_DCAN)

Figure 1-8. Integration Block Diagram for MSS_DCAN Module



1.3.8 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPIA)

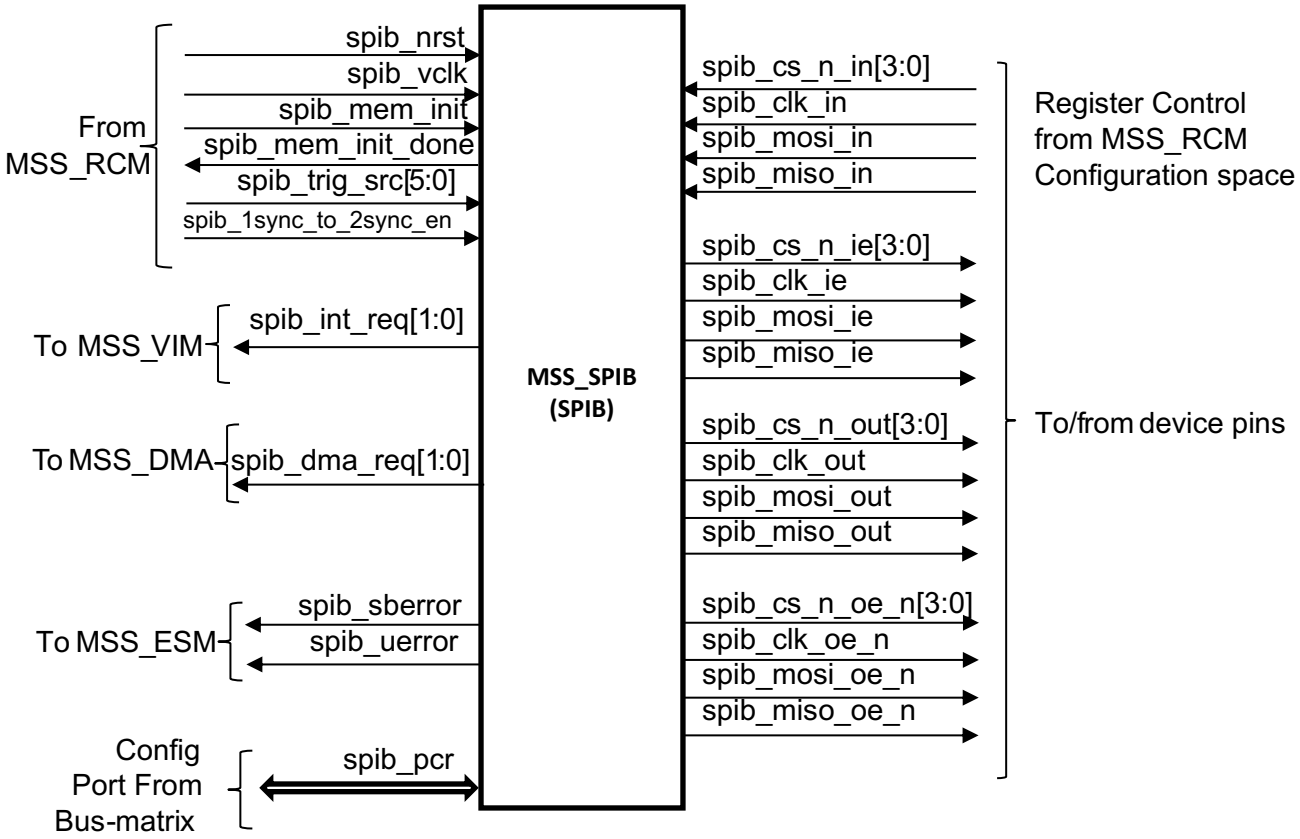
Figure 1-9. MSS_MIBSPIA Integration



1.3.9 Serial Peripheral Interface (MSS_SPIB)

MSS_SPIB module of the 14xx is equivalent in behavior and functionality to MSS_MIBSPIA module configured in compatibility mode for SPI. Users cannot change the compatibility mode configuration of this MSS_SPIB instance.

Figure 1-10. MSS_SPIB Integration Diagram



1.3.10 Quad Serial Peripheral Interface (MSS_QSPI)

The QSPI module of the 14xx only supports one CS pin (qspi1_cs).

1.3.11 Enhanced Direct Memory Access (EDMA)

1.3.11.1 EDMA Controller Integration

The 14xx device has one EDMA channel controller (DSS_TPCC0) on the device:

- DSS_TPCC0 (EDMA TPCC0) has two transfer controllers: DSS_TPTC0 (EDMA TPTC0) and DSS_TPTC1 (EDMA TPTC1)

Table 1-10. DSS_TPCC Configuration

	DSS_TPCC0
Number of DMA channels	64
Number of PaRAM entires	128
Number of QDMA channels	8
Number of event queues	2
Memory protection existence	No
Channel mapping	No
Number of TCs (transfer controllers)	2

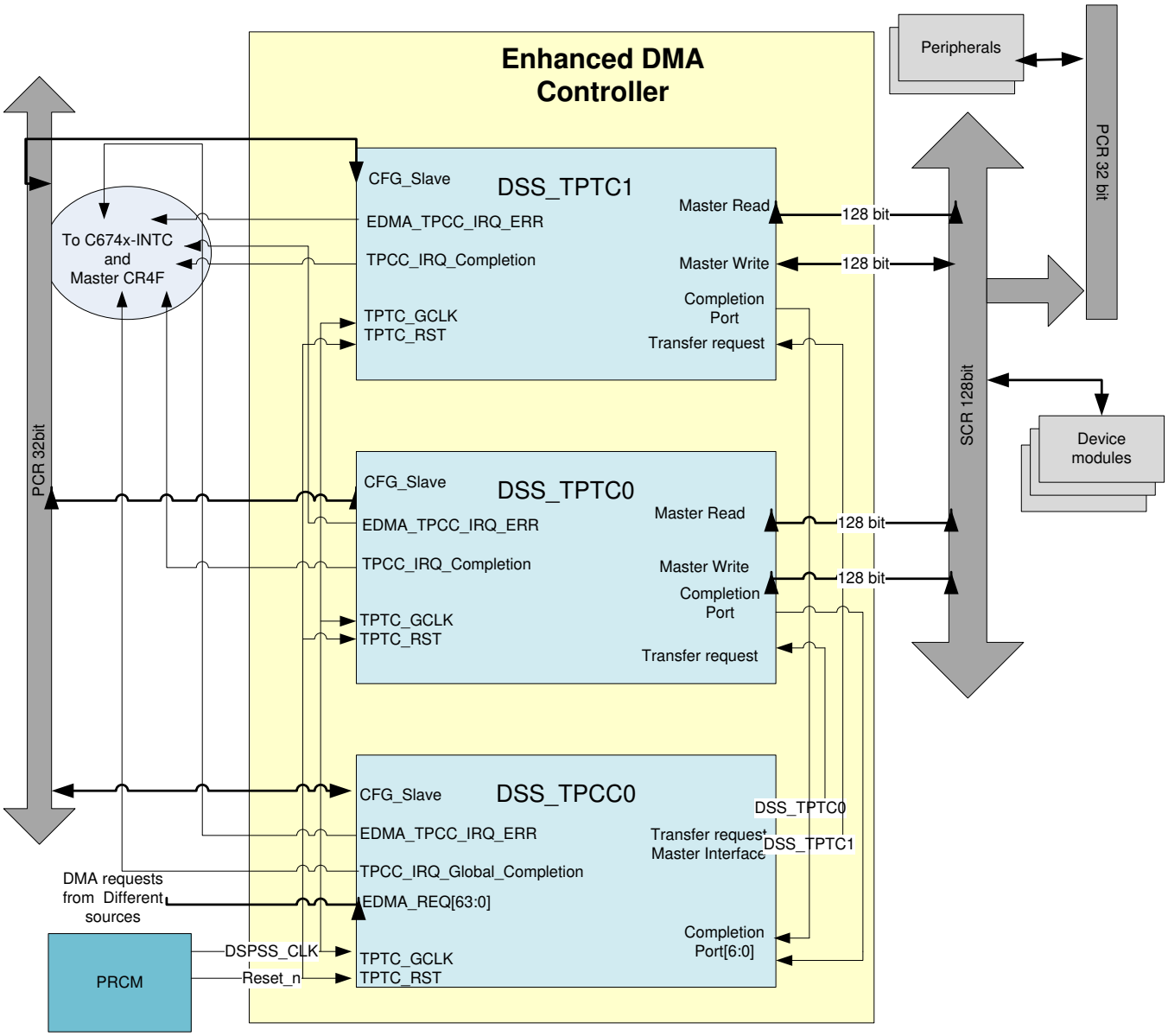
Table 1-11. DSS_TPTC Configuration

	DSS_TPTC[0-1]
FIFO size	512 bytes
TR pipe depth	2
Bus width	16 bytes

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 3-13 and Figure 3-14 show the EDMA controller integration.

Figure 1-11. EDMA Controller Integration



1.3.11.2 EDMA Request Map

Table 1-12. EDMA Request Map

DSS_TPCC0 DMA Req No.	Hardware Event (14XX)
0	DSS_CBUFF_DMA_REQ_0
1	DSS_CBUFF_DMA_REQ_1
2	DSS_CBUFF_DMA_REQ_2
3	DSS_CBUFF_DMA_REQ_3
4	DSS_CBUFF_DMA_REQ_4
5	DSS_CBUFF_DMA_REQ_5
6	DSS_CBUFF_DMA_REQ_6
7	RESERVED
8	Frame Start
9	Chirp Available
10	CSI-2 DMA Req 0
11	CSI-2 DMA Req 1
12	CSI-2 DMA Req 2
13	CSI-2 DMA Req 3
14	VIN_FRAME_START
15	VIN_CHIRP_AVIALABLE
16	VIN_CHANNEL_AVAILABLE
17	DSS_FFT_ACC_CHANNEL_TRIGGER_0
18	DSS_FFT_ACC_CHANNEL_TRIGGER_1
19	DSS_FFT_ACC_CHANNEL_TRIGGER_2
20	DSS_FFT_ACC_CHANNEL_TRIGGER_3
21	DSS_FFT_ACC_CHANNEL_TRIGGER_4
22	DSS_FFT_ACC_CHANNEL_TRIGGER_5
23	DSS_FFT_ACC_CHANNEL_TRIGGER_6
24	DSS_FFT_ACC_CHANNEL_TRIGGER_7
25	DSS_FFT_ACC_CHANNEL_TRIGGER_8
26	DSS_FFT_ACC_CHANNEL_TRIGGER_9
27	DSS_FFT_ACC_CHANNEL_TRIGGER_10
28	DSS_FFT_ACC_CHANNEL_TRIGGER_11
29	DSS_FFT_ACC_CHANNEL_TRIGGER_12
30	DSS_FFT_ACC_CHANNEL_TRIGGER_13
31	DSS_FFT_ACC_CHANNEL_TRIGGER_14
32	DSS_FFT_ACC_CHANNEL_TRIGGER_15
33	RESERVED
34	RESERVED
35	FRC_EVENT_GEN_0
36	FRC_EVENT_GEN_1
37	FRC_EVENT_GEN_2
38	FRC_EVENT_GEN_3
39	RESERVED
40	LOGICAL_FRAME_START
41	ADC_DATA_VALID_FALL
42	RESERVED
43	RESERVED
44	RESERVED
45	RESERVED

Table 1-12. EDMA Request Map (continued)

DSS_TPCC0 DMA Req No.	Hardware Event (14XX)
46	RESERVED
47	RESERVED
48	RESERVED
49	RESERVED
50	RESERVED
51	RESERVED
52	RESERVED
53	RESERVED
54	RESERVED
55	RESERVED
56	RESERVED
57	RESERVED
58	RESERVED
59	RESERVED
60	RESERVED
61	RESERVED
62	RESERVED
63	RESERVED

1.3.12 Error Signaling Module (MSS_ESM)

The 14xx device has one instance of the Error Signaling Module (MSS_ESM), shown in Figure 1-12.

Figure 1-12. 14xx MSS_ESM Integration Diagram

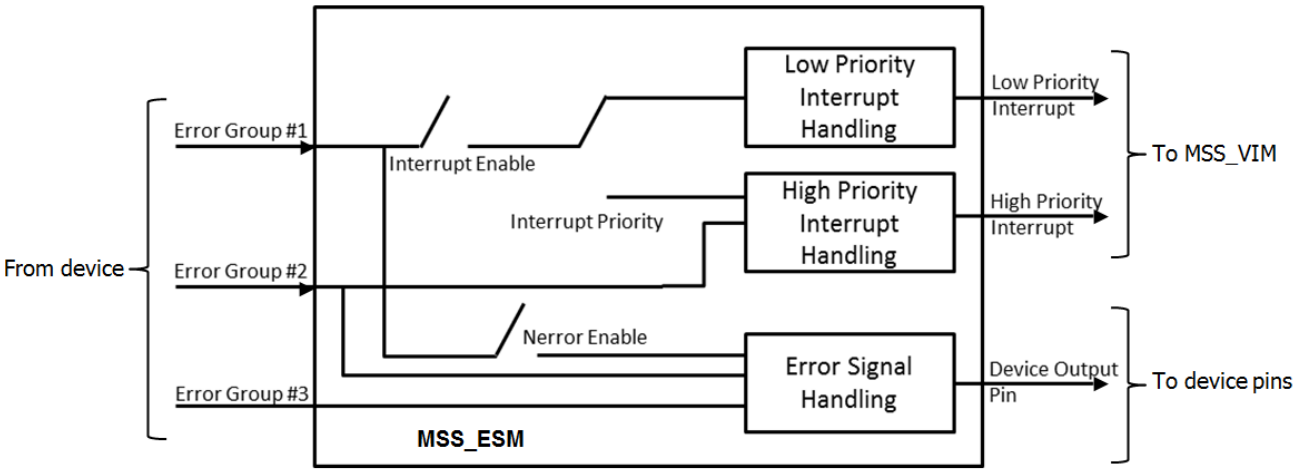


Table 1-13 shows the mapping on the input error inputs to the ESM module from various error sources available for hardware diagnostics within the device.

Table 1-13. MSS_ESM Mapping

ESM Group 1		Channel Type	Description
63	ANA_LIMP_MODE	Error Signal	Error signal at device boot-up, if the CLK monitor finds the REF CLK to be outside the permissible range of frequency
62	MSS_DCCB_ERR	Error Signal	MSS_DCCB frequency comparison error
61	BSS2MSS_MB_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2MSS

Table 1-13. MSS_ESM Mapping (continued)

ESM Group 1		Channel Type	Description
60	BSS2MSS_MB_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_BSS2MSS
59	MSS2BSS_MB_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2BSS
58	MSS2BSS_MB_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2BSS
57	BSS_CRITICAL_ERR	Error Signal	BSS critical Error Indication. Mask control to be configured in IRQ_CTL for individual error signals.
56	Reserved	Reserved	Reserved
55	CLOCK_SUPPLY_ERR	Error Signal	Clock and Supply Errors from Analog. Mask control to be configured in IRQ_CTL for individual error signals
54	Reserved		Reserved
53	NU		
52	NU		
51	NU		
50	NU		
49	NU		
48	MSS_MCRC_ERR	Error Signal	MSS_MCRC Comparison Error
47	NU		
46	NU		
45	NU		
44	Reserved	Reserved	Reserved
43	NU		
42	NU		
41	NU		
40	Reserved	Reserved	Reserved
39	Reserved	Reserved	
38	DSS_CBUFF_SAFETY	Error Signal	CHIRP ERROR or CRC ERROR from DSS_CBUFF
37	NU		
36	DSS_TPTC1_WR_MPU_ERR	Error Signal	DSS_TPTC1 write port MPU error
35	DSS_TPTC1_RD_MPU_ERR	Error Signal	DSS_TPTC1 read port MPU error
34	HVMODE_ERR	Error Signal	Error indication from IO Supply (Supply detector for dual-voltage IOs)
33	MSS_DCAN_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_DCAN Message RAM (FRAM/SRAM)
32	MSS_TCMA_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for Cortex R4F MSS_TCMA
31	Reserved	Reserved	Reserved
30	MSS_DCCA_ERR	Error Signal	MSS_DCCA frequency comparison error
29	DSS_TPTC0_WR_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
28	MSS_TCMB1_REPAIR_ERR	Alert Signal	Single-bit error/reserved indication for MSS_TCMB1
27	MSS_STC_ERR	Error Signal	MSS_STC Error indication for MSS Cortex R4F
26	MSS_TCMB0_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_TCMB0
25	MSS_MIBSPIA_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MIBSPIA multi-buffer (RXRAM/TXRAM)
24	NU		
23	NU		
22	FRC_COMPARE_ERR	Error Signal	Lockstep comparison error from Free running Counter (FRC) in BSS
21	MSS_DCAN_RAM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_DCAN Message Memory (FRAM/SRAM)
20	MSS_VIM_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_VIM_RAM

Table 1-13. MSS_ESM Mapping (continued)

ESM Group 1		Channel Type	Description
19	Reserved		
18	DSS_TPTC0_RD_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
17	MSS_MIBSPIA_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MIBSPIA (RXRAM/TXRAM)
16	NU		
15	MSS_VIM_RAM_FATAL_ERR	Error Signal	Multi-bit uncorrectable error indication for MSS_VIM_RAM
14	NU		
13	Reserved		
12	NU		
11	NU		
10	NU		
9	DSS_CBUFF_ECC_FATAL	Error Signal	Multi-bit error indication from DSS_CBUFF FIFO
8	DSS_CBUFF_ECC_REPAIR	Alert Signal	Single-bit repair indication from DSS_CBUFF FIFO
7	DSS_TPCC_PARITY_ERR	Error Signal	Parity error from DSS_TPCC (EDMA Channel Controller)
6	DSS_CSI_PARITY_ERR	Error Signal	Parity error from DSS_SI
5	NU		
4	MSS_CCCB_ERR	Error Signal	MSS_CCCB(Clock compare core) frequency comparison error
3	MSS_DMA_MEM_PARITY_ERR	Error Signal	Parity Error for DMA1 memory
2	MSS_DMA_MPU_ERR	Error Signal	Error indication from MPU of MSS _DMA
1	MSS_CCCA_ERR	Error Signal	MSS_CCCA(Clock compare core) frequency comparison error
0	NERROR_PAD_IN	Error Signal	Nerror from PAD looped in
ESM Group 2			
31:26	NU		
25	Reserved	Reserved	Reserved
24	MSS_RTIB_NMI	Error Signal	Watchdog Non-mask able interrupt
23:17	NU		
16	MSS_CR4F_LIVELOCK_ERR	Error Signal	Cortex R4F Live lock error
15:9	NU		
8	MSS_TCMB1_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB1
7	NU		
6	MSS_TCMB0_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB0
5	NU		
4	MSS_TCMA_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMA
3	NU		
2	Reserved	Reserved	Reserved
1:0	NU		
ESM Group 3			
31-8	NU		
7	MSS_TCMA_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMA
6	NU		
5	MSS_TCMB1_FATAL_ERR	Error Signal	Multi -bit error indication for MSS_TCMB1
4	NU		
3	MSS_TCMB0_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMB0
2	NU		
1	EFC_AUTOLOAD_ERR	Error Signal	Efuse Auto-load error
0	NU		

1.3.13 High-Speed Interface (HSI)

Table 1-14 lists the high-speed interfaces available for the 14xx device.

Table 1-14. 14xx High-Speed Interfaces

CSI2	4 lanes
LVDS	4 lanes

1.3.14 Handshake RAM (HSRAM)

On the 14xx, there is no HSRAM available.

16xx/18xx

Topic **Page**

2.1 16xx Introduction **162**
2.2 16xx Memory Map **166**
2.3 16xx Integration **173**
2.4 18xx Introduction **203**
2.5 18xx Memory Map **207**
2.6 18xx Integration **215**

2.1 16xx Introduction

2.1.1 16xx Overview

The 16xx is highly integrated single-chip RADAR device in TI's 45-nm low-power RFCMOS technology, a FCBGA 0.65-mm pitch automotive-grade package.

2.1.1.1 Features

- Frequency-Modulated Continuous Wave Radio Frequency Transceiver With 76-81-GHz Band
- Supports Two Transmitter Chains and Four Receiver Chains
- Chirp Profiles With Programmable Period and Slope
- 40-MHz, 50-MHz, 80-MHz, and 100-MHz XTAL/OSC Reference Input Clock
- 12, 14, and 16-bit Real/Complex ADC With Variable Baseband ADC Sampling Rates up to 18.75 MHz at 12-bits Complex
- Cortex R4F at 200-MHz Application Processor for Control Functionality and Safety-Critical Algorithms
- C674x DSP at up to 600 MHz for RADAR Data Processing
- Cortex R4F– Radio Processor at 200 MHz for Continuous Monitoring and Calibration of Analog/RF Functionality
- On-Chip Multicore Debug Support
- Customer-Programmable Efuse Support
- Up to 768KB of L3 Shared Memory Support
- High-Performance Data Transfer With Multiple DMA and EDMA-TPCC Engines
- CAN Support for ECU Interface
- QSPI Serial Flash Support
- MIBSPI, SPI, I2C, and UART Serial Interfaces Support
- Hardware in Loop (HIL) Support
- Two-Lane Serial LVDS Interface Support
- AES, SHA, PKA, and TRNG Engines for Security
- I2C Serial Interface Support
- 3 EPWM (Three Enhanced Pulse Width Modulator)

2.1.2 16xx Description

2.1.2.1 Block Diagram

Figure 2-17 shows the block diagram of the 16xx device.

Figure 2-1. 16xx Block Diagram

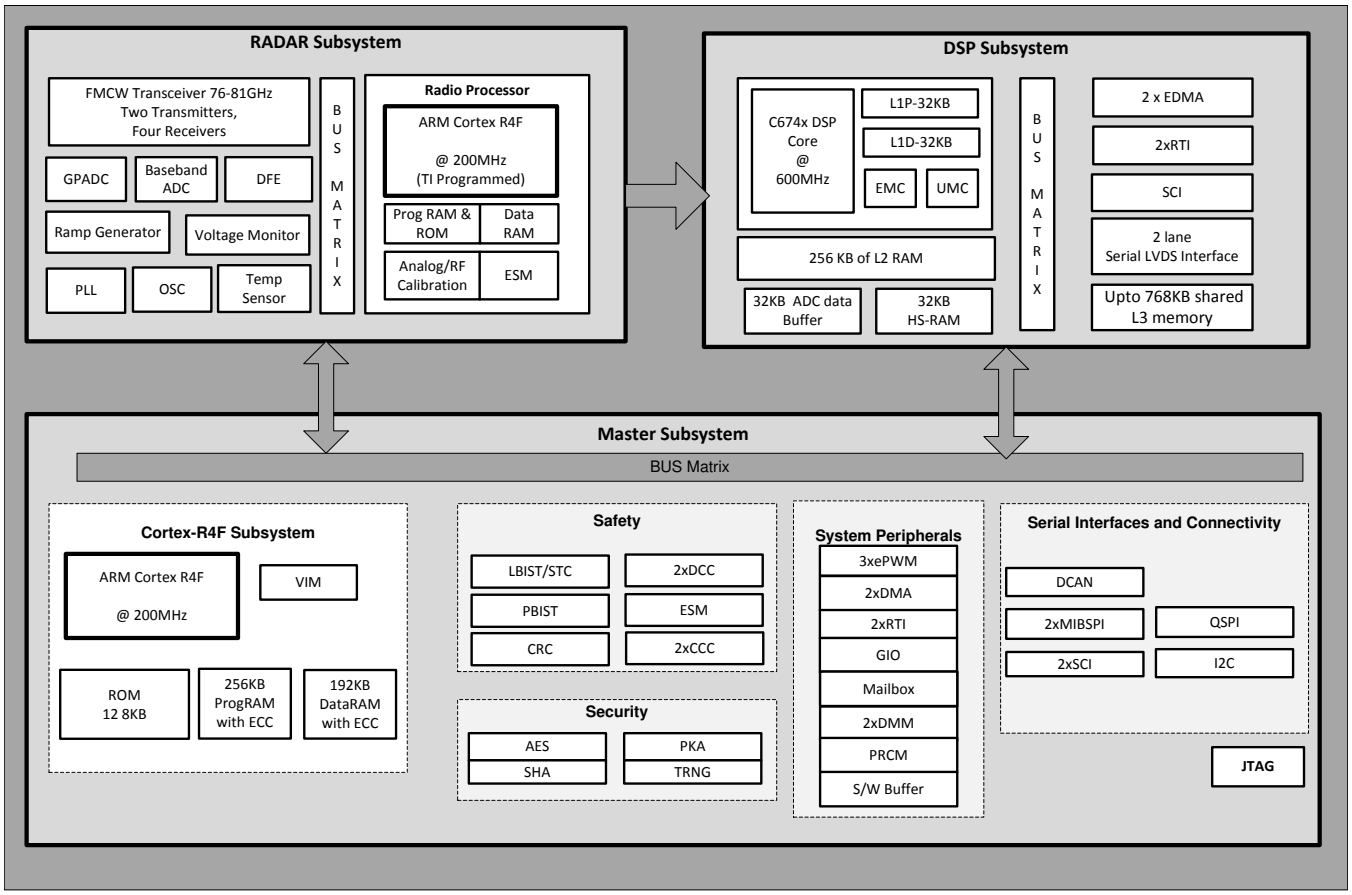


Table 2-1. 16xx Acronyms

ADC	Analog-to-Digital Convertor
AES	Advanced Encryption Standard
CRC	Cyclic Redundancy Check
DCAN	Controller Area Network
DCC	Digital Clock Comparator
DFE	Digital Front End
DMA	Direct Memory Access
DMM	Data Modification Module
ECC	Error Correcting Code
EDMA	Enhanced Direct Memory Access
EMC	Extended Memory Controller
ePWM	Enhanced Pulse Width Modulator
ESM	Error Signaling Module
GIO	General Input/Output
GPADC	General Purpose Analog-to-Digital Convertor
HS-RAM	Handshake RAM
I2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
L1D	Level 1 Data Memory Controller
L1P	Level 1 Program Memory Controller

Table 2-1. 16xx Acronyms (continued)

L2	Level 2
L3	Level 3
LBIST	Logic Built-In Self-Test
LVDS	Low Voltage Differential Signaling
MIBSPI	Multi-Buffered Serial Peripheral Interface Module
OSC	Oscillator
PBIST	Programmable Built-In Self-Test
PKA	Public Key Algorithm
PLL	Phase Locked Loops
PRCM	Power, Reset, Clock Management
QSPI	Quad Serial Peripheral Interface
RTI	Real Time Interrupt
S/W Buffer	Software Buffer
SCI	Serial Communication Interface
SHA	Secure Hash Algorithm
STC	Self-Test Controller
TRNG	True Random Number Generator
UMC	Unified Memory Controller
VIM	Vectored Interrupt Manager

2.1.2.2 Radar Subsystem

The RADAR subsystem is responsible for the RF and analog functionality of the device. The subsystem incorporates a built-in self-test processor for the continuous motoring and calibration of the analog and RF modules. The subsystem consists of:

- FMCW transceiver
  - Integrated PLL, transmitter, receiver, baseband, and A2D
  - 76-81-GHz coverage with 4-GHz available bandwidth
  - Four receive channels
  - Two transmit channels
  - Ultra-accurate chirp engine based on fractional-n PLL
  - 12,14, or 16-bit complex analog to digital converter
- Radio processor for built-in calibration and self-test
  - ARM Cortex R4F-based radio control system
  - Built-in firmware (ROM)
  - Self-calibrating system across frequency and temperature

This subsystem is TI-programmed with an API interface to the on-chip Cortex-R4F application processor.

2.1.2.3 DSP Subsystem

The DSP subsystem consists of the following:

- The TMS320C674x™ VLIW DSP core from the generation, and the TMS320C64x+™ DSP architecture for RADAR data processing. These are enhancements from TI's C64x+™ DSP architecture, with additional features.
- 32KB L1D and 32KB of L1P cache/RAM
- 256KB of L2 RAM
- On-chip L3 shared memory of 768KB, with 256KB dedicated to DSP and 512 KB of memory shared between the DSP and master subsystems.

- 32KB of memory for storing ADC samples from the RADAR subsystem
- Multiple Enhanced Direct Memory Access (EDMA) engines – TPCCs for high-performance data transfers
- 2-lane LVDS interface with support of up to 900 Mbps per lane for the RADAR raw ADC data transfer
- One watchdog timer and a general purpose timer implemented by the real time interrupt (RTI) modules
- One serial communication Interface (SCI) module implementing standard universal asynchronous receiver-transmitter (UART).
- Emulation capabilities
- Little Endian

2.1.2.4 Master Subsystem

The master subsystem consists of the following features:

- Cortex-R4F core supporting ARMv7-R, VFPv3-D16, and ARMv7 debug architecture
- Tightly-coupled memories
  - 128 KB of ROM
  - 256KB of program RAM with ECC
  - 192KB of data RAM with ECC
- Hardware auto-initialization of the memories
- Vectored interrupt manager for prioritizing and controlling the interrupts for different sources

2.1.2.4.1 Serial Interfaces

- One DCAN controller supporting bit rates of up to 1 Mbit/s, and compliant to the controller area network (CAN) 2.0B protocol specification
- One I2C controller module with rates up to 400 kbps
- Two MIBSPI modules
- Two serial communication interface (SCI) modules implementing standard universal asynchronous receiver-transmitter (UART) with baud rates of up to 3.125 Mbps
- One quad SPI module support with maximum rate of 40 MHz

2.1.2.4.2 System Peripherals

- Multiple general-purpose input/output (GPIO) modules
- Direct memory access modules for high-performance data transfers
- One watchdog timer and a general purpose timer implemented by the real-time interrupt (RTI) modules
- Mailbox module for interprocessor communication
- Two data modification modules (DMM) with up to 65 Mbit/s data rate per pin
- Three enhanced pulse width (ePWM) modulator modules
- System reset and control module, which contains registers for the following functions:
  - Status
  - Efuse logic
  - I/O configuration
  - PAD configuration
  - System boot decoding logic

2.1.2.5 Functional Safety Deliverables

See the Device Safety Manual for supported features.

2.1.2.6 On-Chip Debug Support

The on-chip debug support has the following features:

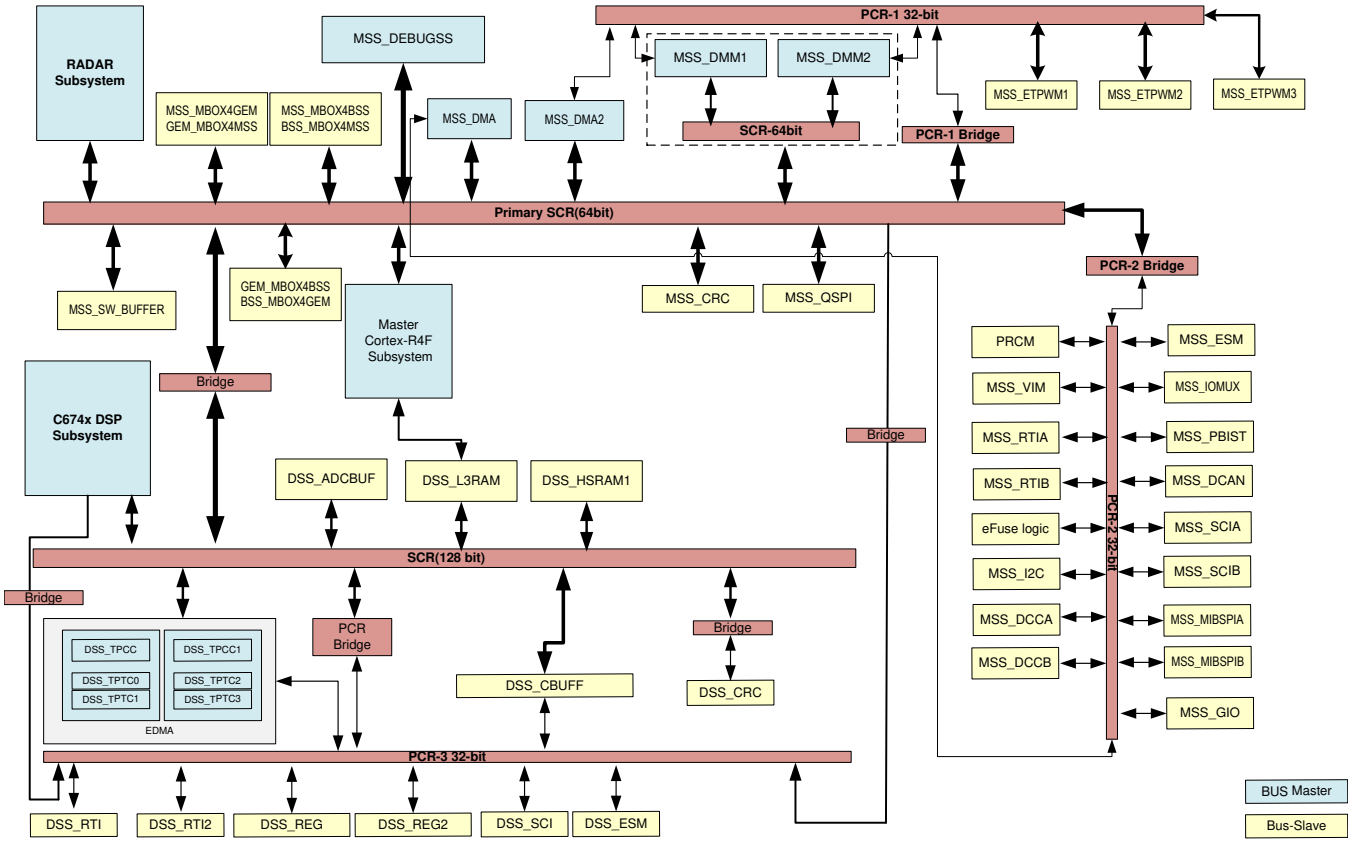
- Multiprocessor debugging to let users control multiple cores embedded in the device, such as:
  - Global starting and stopping of individual or multiple processors
  - Each processor can generate triggers to alter the execution flow of other processors
  - Interconnection of multiple devices
  - Channel triggering
- The following device cores can be debugged through Code Composer Studio (CCS):
  - Cortex-R4F
  - DSP
- Target debugging using IEEE1149.1 (JTAG®) port
- The debug subsystem includes:
  - IEEE1149.7 adapter
  - Generic TAP for emulation and test control (ICEPick-D™)
  - Debug access port (DAP)
  - Embedded trace macro (ETM)
  - Trace port interface Unit (TPIU)
  - Embedded trace buffer (ETB)

2.2 16xx Memory Map

2.2.1 System Interconnect

The device implements a system interconnect based on TI's common bus architecture, comprising of VBUSM and VBUSP protocols. Figure 2-18 shows the interconnect diagram.

Figure 2-2. System Interconnect



The system interconnect is designed for the high-performance needs of the system. Its divided into three interconnect systems local to each of the three subsystems: the RADAR subsystem, DSP subsystem, and master subsystem. The interconnection of all these subsystems is shown in Figure 3-2.

In the master subsystem, the primary VBUSM SCR is responsible for managing the arbitration priority between accesses from multiple masters to each of the slaves. The arbitration priority is always round-robin.

The master subsystem has PCR interconnect that manages the accesses to the peripheral registers and peripheral memories, and provides a global reset for all peripherals. It also supports the capability to selectively enable or disable the clock for each peripheral individually. The PCR also manages the accesses to the system module registers required to configure the device clocks, interrupts, and so forth. The system module registers include status flags for indicating exception conditions – resets, aborts, errors, and interrupts.

Similarly, the 128-bit VBUSM SCR in the DSP subsystem manages the arbitration between accesses from the multiple masters to the slaves. The DSP subsystem has a 32-bit VBUSP PCR for the system and non-system peripherals.

2.2.2 Master Subsystem Cortex-R4F Memory Map

Table 3-2 shows the master subsystem, Cortex-R4F memory map.

Table 2-2. Master Subsystem, Cortex-R4F Memory Map

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_TCMA_ROM	0x0000_0000	0x0001_7FFF	128KiB	MSS_TCMA_ROM (TCMA) Program ROM (refer to ROM Eclipsing section)
Reserved	0x0001_8000	0x001F_FFFF		Reserved (refer to ROM Eclipsing section)

Table 2-2. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_TCMA_RAM	0x0020_0000	0x07FF_FFFF	256KiB/ 512KiB	MSS_TCMA_RAM (TCMA) size varies based on device and DSS_L3 (L3) sharing options configured (refer to ROM Eclipsing section)
MSS_TCMB	0x0800_0000	0x0C1F_FFFF	192KiB	MSS_TCMB (TCMB) Data RAM
MSS_SW_BUFFER	0x0C20_0000	0x0C20_1FFF	8KiB	MSS_SW_BUFFER (SWBUFF) Scratchpad memory
Reserved	0x0C20_2000	0x4FFF_FFFF		Reserved
DSS_TPTC0	0x5000_0000	0x5000_03FF	792B	DSS_TPTC0 (EDMA TPTC0) module configuration space
DSS_REG	0x5000_0400	0x5000_07FF	864B	DSS_REG (DSPSS) control module registers
DSS_TPTC1	0x5000_0800	0x5000_0BFF	792B	DSS_TPTC1 (EDMA TPTC1) module configuration space
DSS_REG2	0x5000_0C00	0x5000_FFFF	676B	DSS_REG2 (DSPSS) control module registers
DSS_TPCC	0x5001_0000	0x5001_FFFF	16KiB	DSS_TPCC (EDMA TPCC0) module configuration space
DSS_RTI	0x5002_0000	0x5002_FFFF	192B	DSS_RTI (WDT/RTI1) configuration space
DSS_SCI	0x5003_0000	0x5003_FFFF	148B	DSS_SCI (SCI) memory space
DSS_STC	0x5004_0000	0x5004_FFFF	284B	DSS_STC (STC) module configuration space (refer to Safety chapter)
Reserved	0x5005_0000	0x5006_FFFF		Reserved
DSS_CBUFF	0x5007_0000	0x5007_FFFF	564B	DSS_CBUFF (CBUFF) module configuration registers (refer to HSI chapter)
Reserved	0x5008_0000	0x5008_FFFF		Reserved
DSS_TPTC2	0x5009_0000	0x5009_03FF	792B	DSS_TPTC2 (EDMA TPTC2) module configuration space
DSS_TPTC3	0x5009_0400	0x5009_FFFF	792B	DSS_TPTC3 (EDMA TPTC3) module configuration space
DSS_TPCC1	0x500A_0000	0x500A_FFFF	16KiB	DSS_TPCC1 (EDMA TPCC1) module configuration space
Reserved	0x500B_0000	0x500C_FFFF		Reserved
DSS_ESM	0x500D_0000	0x500E_FFFF	1KiB	DSS_ESM (ESM) module configuration registers (refer to Safety chapter)
DSS_RTI2	0x500F_0000	0x500F_FFFF	192B	DSS_RTI2 (RTI2) module configuration registers
Reserved	0x5010_0000	0x50FF_FFFF		Reserved
DSS_L3RAM	0x5100_0000	0x51FF_FFFF	2MB	DSS_L3RAM (L3) shared memory space
DSS_ADCBUF	0x5200_0000	0x5201_FFFF	32KiB	DSS_ADCBUF (ADC) buffer memory space
DSS_CBUFF_FIFO	0x5202_0000	0x5202_7FFF	16KiB	DSS_CBUFF_FIFO (CBUFF) FIFO space (refer to HSI chapter)
Reserved	0x5202_8000	0x5207_FFFF		Reserved
DSS_HSRAM1	0x5208_0000	0x5208_FFFF	32KiB	DSS_HSRAM1 (HSRAM) Handshake memory space
Reserved	0x5209_0000	0x577D_FFFF		Reserved
DSS_DSP_L2_UMAP1	0x577E_0000	0x577F_FFFF	128KiB	DSS_DSP_L2_UMAP1 (L2) RAM space
DSS_DSP_L2_UMAP0	0x5780_0000	0x57DF_FFFF	128KiB	DSS_DSP_L2_UMAP0 (L2) RAM space
DSS_DSP_L1P	0x57E0_0000	0x57EF_FFFF	32KiB	DSS_DSP_L1P (L1) program memory space
DSS_DSP_L1D	0x57F0_0000	0xBFFF_FFFF	32KiB	DSS_DSP_L1D (L1) data memory space

Table 2-2. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
EXT_FLASH	0xC000_0000	0xC07F_FFFF	8MB	MSS_QSPI (QSPI) flash memory space
MSS_QSPI	0xC080_0000	0xF060_0FFF	116B	MSS_QSPI (QSPI) module configuration registers
MSS_MBOX4BSS	0xF060_1000	0xF060_1FFF	2KiB	MSS_MBOX4BSS mailbox memory space
BSS_MBOX4MSS	0xF060_2000	0xF060_3FFF	2KiB	BSS_MBOX4MSS mailbox memory space
GEM_MBOX4MSS	0xF060_4000	0xF060_4FFF	2KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0xF060_5000	0xF060_5FFF	2KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0xF060_6000	0xF060_6FFF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0xF060_7000	0xF060_7FFF	2KiB	BSS_MBOX4GEM mailbox memory space
BSS_MBOX4MSS_REG	0xF060_8000	0xF060_80FF	188B	BSS_MBOX4MSS_REG mailbox Configuration registers
BSS_MBOX4GEM_REG	0xF060_8100	0xF060_81FF	188B	BSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4BSS_REG	0xF060_8200	0xF060_82FF	188B	GEM_MBOX4BSS_REG mailbox Configuration registers
MSS_MBOX4GEM_REG	0xF060_8300	0xF060_83FF	188B	MSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4MSS_REG	0xF060_8400	0xF060_85FF	188B	GEM_MBOX4MSS_REG mailbox Configuration registers
MSS_MBOX4BSS_REG	0xF060_8600	0xFCF7_8BFF	188B	MSS_MBOX4BSS_REG mailbox Configuration registers
MSS_ETPWM1	0xFCF7_8C00	0xFCF7_8CFF	116B	MSS_ETPWM1 (ePWM1) configuration registers
MSS_ETPWM2	0xFCF7_8D00	0xFCF7_8DFF	116B	MSS_ETPWM2 (ePWM2) configuration registers
MSS_ETPWM3	0xFCF7_8E00	0xFCF8_0FFF	116B	MSS_ETPWM3 (ePWM3) configuration registers
MSS_DMA2_RAM	0xFCF8_1000	0xFCFF_0FFF	4KiB	MSS_DMA2_RAM (DMA2) RAM memory space
MSS_PCR2	0xFCFF_1000	0xFCFF_F5FF	1KiB	MSS_PCR2 (PCR_2) interconnect configuration port
MSS_DMM2	0xFCFF_F600	0xFCFF_F6FF	472B	MSS_DMM2 (DMM2) module configuration registers
MSS_DMM	0xFCFF_F700	0xFCFF_F7FF	472B	MSS_DMM (DMM1) module configuration registers
MSS_DMA2_REG	0xFCFF_F800	0xFCFF_FFFF	1KiB	MSS_DMA2_REG (DMA2) module configuration registers
MSS_DTHE	0xFD00_0000	0xFD00_3FFF	3KiB	MSS_DTHE (Crypto) module configuration registers
Reserved	0xFD00_4000	0xFDFF_FFFF		Reserved
MSS_MCRC	0xFE00_0000	0xFF0B_FFFF	16KiB	MSS_MCRC (CRC) module configuration registers
MSS_MIBSPIB_TXRAM	0xFF0C_0000	0xFF0C_01FF	0.5KiB	MSS_MIBSPIB_TXRAM (MIBSPIB) TX RAM memory space
MSS_MIBSPIB_RXRAM	0xFF0C_0200	0xFF0D_FFFF	0.5KiB	MSS_MIBSPIB_RXRAM (MIBSPIB) RX RAM memory space
MSS_MIBSPIA_TXRAM	0xFF0E_0000	0xFF0E_01FF	0.5KiB	MSS_MIBSPIA_TXRAM (MIBSPIA) TX RAM memory space
MSS_MIBSPIA_RXRAM	0xFF0E_0200	0xFF1D_FFFF	0.5KiB	MSS_MIBSPIA_RXRAM (MIBSPIA) RX RAM memory space
MSS_DCAN_MEM	0xFF1E_0000	0xFF4F_FFFF	128KiB	MSS_DCAN_MEM (CAN) RAM memory space
Reserved	0xFF50_0000	0xFF9F_FFFF	68KiB	Reserved

Table 2-2. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_DEBUGSS	0xFFA0_0000	0xFFF7_7FFF	244KiB	MSS_DEBUGSS (Debug subsystem) memory space and registers
MSS_PCR	0xFFF7_8000	0xFFF7_9FFF	1KiB	MSS_PCR (PCR_1) interconnect configuration port
Reserved	0xFFF7_A000	0xFFF7_BBFF	452B	Reserved
MSS_GIO	0xFFF7_BC00	0xFFF7_C7FF	180B	MSS_GIO (GIO) module configuration registers
Reserved	0xFFF7_C800	0xFFF7_D3FF	768B	Reserved
MSS_I2C	0xFFF7_D400	0xFFF7_DBFF	112B	MSS_I2C (I2C) module configuration registers
MSS_DCAN	0xFFF7_DC00	0xFFF7_E4FF	512B	MSS_DCAN (CAN) module configuration registers
MSS_SCIA	0xFFF7_E500	0xFFF7_E6FF	148B	MSS_SCIA (SCIA/UART) module configuration registers
MSS_SCIB	0xFFF7_E700	0xFFF7_F3FF	148B	MSS_SCIB (SCIB/UART) module configuration registers
MSS_MIBSPIA	0xFFF7_F400	0xFFF7_F5FF	512B	MSS_MIBSPIA (MIBSPIA) module configuration registers
MSS_MIBSPIB	0xFFF7_F600	0xFFF7_FFFF	512B	MSS_MIBSPIB (MIBSPIB) module configuration registers
MSS_DMA_RAM	0xFFF8_0000	0xFFF8_1FFF	4KiB	MSS_DMA_RAM (DMA1) RAM memory space
MSS_VIM_MEM	0xFFF8_2000	0xFFF8_BFFF	2KB	MSS_VIM_MEM (VIM) RAM memory space
Reserved	0xFFF8_C000	0xFFFF_E0FF		Reserved
MSS_TOPRCM	0xFFFF_E100	0xFFFF_E3FF	756B	MSS_TOPRCM TOP Level Reset, Clock management registers
MSS_PBIST	0xFFFF_E400	0xFFFF_E5FF	464B	MSS_PBIST (PBIST) module configuration registers (refer to Safety chapter)
MSS_STC	0xFFFF_E600	0xFFFF_E9FF	284B	MSS_STC (STC) module configuration registers (refer to Safety chapter)
MSS_IOMUX	0xFFFF_EA00	0xFFFF_EBFF	512B	MSS_IOMUX (IOMUX) module registers
MSS_DCCA	0xFFFF_EC00	0xFFFF_EDFF	44B	MSS_DCCA (DCCA) module configuration registers
MSS_RTIB	0xFFFF_EE00	0xFFFF_EFFF	192B	MSS_RTIB (WDT/RTIB) module configuration registers
MSS_DMA_REG	0xFFFF_F000	0xFFFF_F3FF	1KiB	MSS_DMA_REG (DMA1) module configuration registers
MSS_DCCB	0xFFFF_F400	0xFFFF_F4FF	44B	MSS_DCCB (DCCB) module configuration registers
MSS_ESM	0xFFFF_F500	0xFFFF_F5FF	156B	MSS_ESM (ESM) module configuration registers (refer to Safety chapter)
Reserved	0xFFFF_F600	0xFFFF_F7FF		Reserved
MSS_GPCFG_REG	0xFFFF_F800	0xFFFF_FBFF	352B	MSS_GPCFG_REG (GPCFG) General purpose control registers
MSS_RTIA	0xFFFF_FC00	0xFFFF_FCFF	192B	MSS_RTIA (RTIA) module configuration registers
MSS_VIM	0xFFFF_FD00	0xFFFF_FEFF	512B	MSS_VIM (VIM) module configuration registers
MSS_RCM	0xFFFF_FF00	0xFFFF_FFFF	256B	MSS_RCM (RCM) Reset, Clock management registers

2.2.2.1 Radar Subsystem Interface

The RADAR subsystem is accessible through a set of TI-implemented high-level API calls by the application running on the master CR4F. For more information on the 16xx RADAR subsystem interface, see the 16xx Interface control document.

2.2.3 DSP Subsystem Memory Map

Table 3-3 shows the DSP C674x memory map.

Table 2-3. DSP C674x Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSP_L2_UMAP1	0x007E_0000	0x007F_FFFF	128KiB	DSP_L2_UMAP1 (L2) RAM space
DSP_L2_UMAP0	0x0080_0000	0x0081_FFFF	128KiB	DSP_L2_UMAP0 (L2) RAM space
DSP_L1P	0x00E0_0000	0x00E0_7FFF	32KiB	DSP_L1P (L1) program memory space
DSP_L1D	0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D (L1) data memory space
DSS_TPTC0	0x0200_0000	0x0200_03FF	1KiB	DSS_TPTC0 (EDMA TPTC0) module configuration space
DSS_REG	0x0200_0400	0x0200_07FF	864B	DSS_REG (DSPSS) control module registers
DSS_TPTC1	0x0200_0800	0x0200_0BFF	1KiB	DSS_TPTC1 (EDMA TPTC1) module configuration space
DSS_REG2	0x0200_0C00	0x0200_0FFF	624B	DSS_REG2 (DSPSS) control module registers
DSS_TPCC	0x0201_0000	0x0201_3FFF	16KiB	DSS_TPCC (EDMA TPCC0) module configuration space
DSS_RTI	0x0202_0000	0x0202_00FF	192B	DSS_RTI (WDT/RTI1) module configuration registers
DSS_SCI	0x0203_0000	0x0203_00FF	148B	DSS_SCI (SCI/UART) module Configuration registers
DSS_CBUFF	0x0207_0000	0x0207_03FF	564B	DSS_CBUFF (CBUFF) module Configuration registers (refer to HSI chapter)
Reserved	0x0208_0000	0x0208_FFFF		Reserved
DSS_TPTC2	0x0209_0000	0x0209_03FF	1KiB	DSS_TPTC2 (EDMA TPTC2) module configuration space
DSS_TPTC3	0x0209_0400	0x0209_07FF	1KiB	DSS_TPTC3 (EDMA TPTC3) module configuration space
DSS_TPCC1	0x020A_0000	0x020A_3FFF	16KiB	DSS_TPCC1 (EDMA TPCC1) module configuration space
Reserved	0x020B_0000	0x020C_FFFF		Reserved
DSS_ESM	0x020D_0000	0x020E_FFFF	92B	DSS_ESM (ESM) module Configuration registers (refer to Safety chapter)
DSS_RTI2	0x020F_0000	0x020F_00FF	192B	DSS_RTI2 (RTI2) module configuration registers
Reserved	0x0210_0000	0x0460_7FFF		Reserved
BSS_MBOX4MSS_REG	0x0460_8000	0x0460_80FF	188B	BSS_MBOX4MSS_REG mailbox Configuration registers
BSS_MBOX4GEM_REG	0x0460_8100	0x0460_81FF	188B	BSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4BSS_REG	0x0460_8200	0x0460_82FF	188B	GEM_MBOX4BSS_REG mailbox Configuration registers
MSS_MBOX4GEM_REG	0x0460_8300	0x0460_83FF	188B	MSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4MSS_REG	0x0460_8400	0x0460_84FF	188B	GEM_MBOX4MSS_REG mailbox Configuration registers
MSS_MBOX4BSS_REG	0x0460_8600	0x0460_86FF	188B	MSS_MBOX4BSS_REG mailbox Configuration registers
Reserved	0x050C_0000	0x1FFF_FFFF		Reserved
DSS_L3RAM	0x2000_0000	0x201F_FFFF	2MB	DSS_L3RAM (L3) shared memory space

Table 2-3. DSP C674x Memory Map (continued)

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_ADCBUF	0x2100_0000	0x2100_7FFC	32KiB	DSS_ADCBUF (ADC buffer) memory space
DSS_CBUFF_FIFO	0x2102_0000	0x2102_3FFC	16KiB	DSS_CBUFF_FIFO (Common buffer) FIFO space (Refer to HSI chapter)
Reserved	0x2102_8000	0x2107_FFFF		Reserved
DSS_HSRAM1	0x2108_0000	0x2108_7FFC	32KiB	DSS_HSRAM1 (HSRAM) Handshake memory space
Reserved	0x2109_0000	0x21FF_FFFF		Reserved
DSS_MCRC	0x2200_0000	0x2200_03FF	1KiB	DSS_MCRC (CRC) module Configuration registers
Reserved	0x2500_0000	0x5060_0FFF		Reserved
MSS_MBOX4BSS	0x5060_1000	0x5060_17FF	2KiB	MSS_MBOX4BSS mailbox memory space
BSS_MBOX4MSS	0x5060_2000	0x5060_27FF	2KiB	BSS_MBOX4MSS mailbox memory space
GEM_MBOX4MSS	0x5060_4000	0x5060_47FF	2KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0x5060_5000	0x5060_57FF	2KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0x5060_6000	0x5060_67FF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_57FF	2KiB	BSS_MBOX4GEM mailbox memory space
Reserved	0x5600_0000	0x5060_5FFF		Reserved
GEM_MBOX4BSS	0x5060_6000	0x5060_67FF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_7FFF	2KiB	BSS_MBOX4GEM mailbox memory space
Reserved	0x5600_0000	0xFFFF_FFFF		Reserved

2.2.4 EDMA Memory Map

Table 3-4 shows the EDMA-TPTC memory map.

Table 2-4. EDMA-TPTC Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_EDMA_SCI	0x0603_0000	0x0603_00FF	148B	DSS_SCI memory space view from EDMA
DSS_DSP_L2_UMAP1	0x107E_0000	0x107F_FFFF	128KiB	DSP_L2_UMAP1 (L2) memory view from EDMA
DSS_DSP_L2_UMAP0	0x1080_0000	0x1081_FFFF	128KiB	DSP_L2_UMAP0 (L2) memory view from EDMA
DSS_DSP_L1P	0x10E0_0000	0x10E0_7FFF	32KiB	DSP_L1P (L1) program memory view from EDMA
DSS_DSP_L1D	0x10F0_0000	0x10F0_7FFF	32KiB	DSP_L1D (L1) data memory view from EDMA
DSS_L3RAM	0x2000_0000	0x201F_FFFF	2MB	DSS_L3RAM shared memory space
DSS_ADCBUF	0x2100_0000	0x2100_7FFC	32KiB	DSS_ADCBUF memory space
DSS_CBUFF_FIFO	0x2102_0000	0x2102_3FFC	16KiB	DSS_CBUFF_FIFO (Common buffer) memory space (Refer to HSI chapter)
DSS_HSRAM1	0x2108_0000	0x2108_7FFC	32KiB	DSS_HSRAM1 (Handshake) memory
MSS_TCMA_RAM	0x4020_0000	0x4023_FFFF	256 KiB	MSS_TCMA_RAM (TCMA) Data RAM
MSS_TCMB	0x4800_0000	0x4802_FFFF	192 KiB	MSS_TCMB (TCMB) Data RAM
MSS_SW_BUFFER	0x4C20_0000	0x4C20_1FFF	8 KiB	MSS_SW_BUFFER S/W Scratchpad memory
GEM_MBOX4MSS	0x5060_4000	0x5060_4000	2 KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0x5060_5000	0x5060_5000	2 KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0x5060_6000	0x5060_6000	2 KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_7000	2 KiB	BSS_MBOX4GEM mailbox memory space

2.3 16xx Integration

2.3.1 Cortex-R4F Subsystem

2.3.1.1 Tightly Coupled Memories

Table 3-5 lists the dedicated MSS_TCMA_RAM and MSS_TCMB sizes for the Cortex R4F processor in the master subsystem, and also mentions the total available L3 shared RAM in the device. A portion of this L3 shared memory (DSS_L3RAM) can be allotted as TCM, to further increase the MSS_TCMA_RAM and MSS_TCMB available for the Cortex R4F.

Table 2-5. TCM and Shared Memory Available for Cortex R4F in Master Subsystem

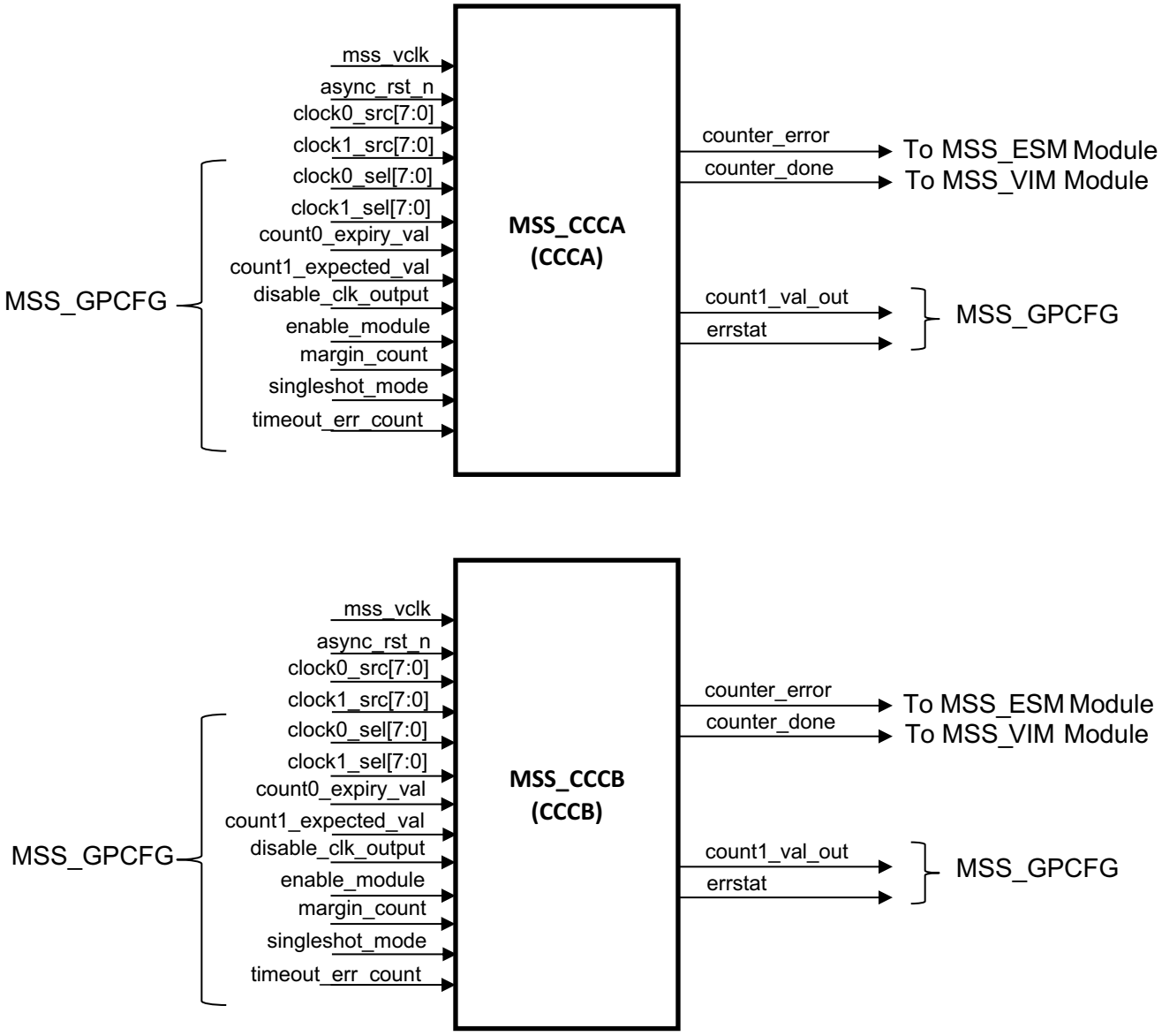
Cortex R4F		Shared
MSS_TCMA_RAM (Program RAM) (KB)	MSS_TCMB (Data RAM) (KB)	L3 Shared (KB)
256	192	768

See on how the L3 shared memory (DSS_L3RAM) can be assigned between the Cortex R4F of the master subsystem and the DSP core.

2.3.2 Clock Comparator

2.3.2.1 Core Clock Comparator (MSS_CCCA/MSS_CCCB)

Figure 2-3. Integration of MSS_CCCA and MSS_CCCB Modules



2.3.2.1.1 MSS_CCCA and MSS_CCCB Integration Connections

This device has two instances of CCC: MSS_CCCA (CCCA) and MSS_CCCB (CCCB). The clock connectivity information for these two instances is provided in Table 3-6. Configuration and status of this module is available through the MSS_GPCFG registers of the device.

Table 2-6. MSS_CCCA and MSS_CCCB Integration Connections

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
counter_error	ESM_GRP1[1]	ESM_GRP1[4]
counter_done	IRQ[80]	IRQ[81]

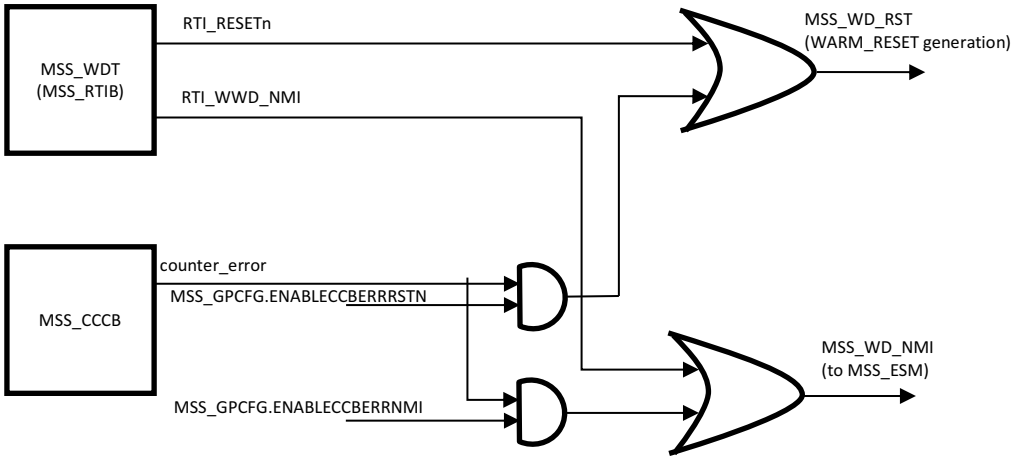
Table 2-6. MSS_CCCA and MSS_CCCB Integration Connections (continued)

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
clock0_src[0]	REFCLK	CR4_VCLK
clock0_src[1]	CPUCLK	DSSCLK
clock0_src[2]	RCCLK	BSSCLK
clock0_src[3]	RCCLK	QSPICLK
clock0_src[4]	RCCLK	N/A
clock0_src[5]	RCCLK	REFCLK
clock0_src[6]	RCCLK	CPUCLK
clock0_src[7]	RCCLK	RCCLK
clock1_src[0]	REFCLK	PLLCLK_600
clock1_src[1]	PLLCLK_600	MSS_VCLK
clock1_src[2]	PLL 240Mhz	CPUCLK
clock1_src[3]	RCCLK	CR4_VCLK
clock1_src[4]	RCCLK	MSS_VCLK
clock1_src[5]	RCCLK	DSSCLK
clock1_src[6]	RCCLK	BSSCLK
clock1_src[7]	RCCLK	QSPICLK

2.3.2.1.2 MSS_CCCB Integration to MSS_WD

As a safety requirement, WDT IP should work on an independent clock source instead of the clock used by the MSS CR4. Because the WDT IP does not allow this flexibility, an additional monitoring logic is added in the form of CCM (MSS_CCCB instance), coupled along with the watchdog. MSS_CCCB is used to compare CR4_VCLK to an independent reference clock, such as XTAL. If the CR4 clock indicates a deviation from the expected frequency, a WD reset or a WD NMI can be issued.

Figure 2-4. MSS_CCCB Integration to MSS_WD



2.3.2.2 Dual Clock Comparator (MSS_DCCA/MSS_DCCB)

Figure 2-5. MSS_DCCA/MSS_DCCB Integration Diagram

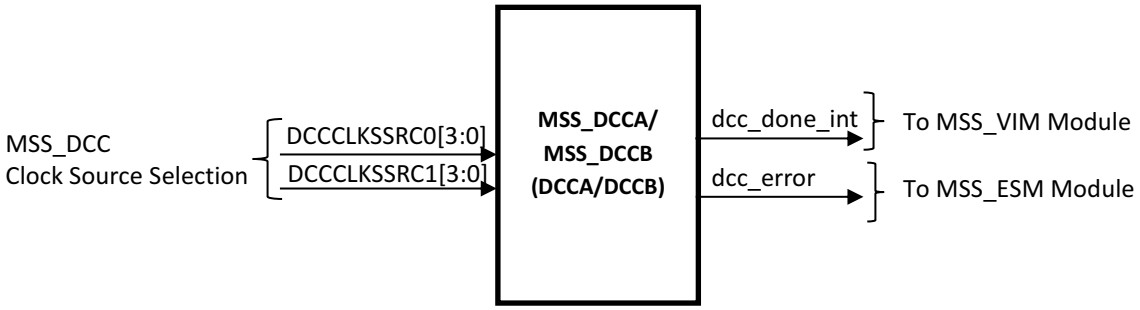


Table 2-7. MSS_DCCA Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - REF_CLK	0x0 - REF_CLK
0xA - PLL_600	0x1 - CPU_CLK
0x5 - PLL_240	0x2 to 0x7 - RC_CLK

Table 2-8. MSS_DCCB Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - PLL_600	0x0 - VCLK
0xA - VCLK	0x1 - DSS_CLK
0x5 - CPU_CLK	0x2 - BSS_CLK
	0x3 - QSPI_CLK
	0x4 - Reserved
	0x5 - REF_CLK
	0x6 - CPU_CLK
	0x7 - RC_CLK

NOTE: Any values not mentioned are not used.

2.3.3 C674x DSP Subsystem

2.3.3.1 DSP Event Assignment

Table 2-9. DSP Event Assignment

Event No.	Interrupt	Description
0	EVT0	Output of event combiner0, for events 1 through 31
1	EVT1	Output of event combiner0, for events 32 through 63
2	EVT2	Output of event combiner0, for events 64 through 95
3	EVT3	Output of event combiner0, for events 96 through 127
4	Reserved	Reserved
5	Reserved	Reserved
6	Reserved	Reserved
7	Reserved	Reserved
8	Reserved	Reserved
9	Reserved	Reserved

Table 2-9. DSP Event Assignment (continued)

Event No.	Interrupt	Description
10	Reserved	Reserved
11	Reserved	Reserved
12	Reserved	Reserved
13	IDMAINT0	From DSP EMC, IDMA Channel 0 Interrupt
14	IDMAINT1	From DSP EMC, IDMA Channel 1Interrupt
15	Reserved	Reserved
16	DSS_TPTC0_IRQ_DONE	DSS_TPTC0 (EDMA TPTC0) completion interrupt
17	DSS_TPTC0_IRQ_ERR	DSS_TPTC0 (EDMA TPTC0) Error Interrupt
18	DSS_TPTC1_IRQ_DONE	DSS_TPTC1 (EDMA TPTC1) completion interrupt
19	DSS_TPTC1_IRQ_ERR	DSS_TPTC1 (EDMA TPTC1) Error Interrupt
20	DSS_TPCC_IRQ_DONE	DSS_TPCC (EDMA TPCC0) Global completion Interrupt
21	DSS_TPCC_IRQ_ERR	DSS_TPCC (EDMA TPCC0) Error Interrupt
22	DSS_CBUFF_IRQ	DSS_CBUFF (COMMON BUFFER) Interrupt
23	Reserved	Reserved
24	DSS_CBUFF_ERR_INTR	DSS_CBUFF (COMMON BUFFER) Error Interrupt
25	Reserved	Reserved
26	DSS_FRAME_START_IRQ/DSS_DMMSWIN T0/DSS_DMMSWINT39	Mux of VIN Frame start or BSS DFE Frame start.
27	DSS_CHIRP_AVAIL_IRQ/DSS_DMMSWINT2 /DSS_DMMSWINT43	Mux of VIN Chirp Available or DFE chirp available..
28	Reserved	Reserved
29	Reserved	Reserved
30	Reserved	Reserved
31	Reserved	Reserved
32	DSS_ESM_LOW_PRIORITY	MSS_ESM_IRQ (Aggregate of MSS_ESM_GP1 errors)
33	DSS_MCRC_INT	MSS_MCRC (CRC) Interrupt
34	DSS_PROG_FILT_ERR	Error interrupt from Programmable filter indicating wrong programming of filter length exceeding the allowed range.
35	GEM_WAKEUP_SOURCE_FROM_DFT	Wakeup source from DFT module.
36	DSS_STC_DONE	Done indication from DSS_STC
37	DSP_PBIST_DONE	DSP_PBIST done indication from GEM
38	Reserved	Reserved
39	Reserved	Reserved
40	Reserved	Reserved
41	Reserved	Reserved
42	Reserved	Reserved
43	Reserved	Reserved
44	Reserved	Reserved
45	Reserved	Reserved
46	DSS_DMMSWINT8	Interrupt from DSS_DMM configurable
47	DSS_DMMSWINT4	Interrupt from DSS_DMM configurable
48	Reserved	Reserved
49	Reserved	Reserved
50	Reserved	Reserved
51	Reserved	Reserved
52	Reserved	Reserved
53	Reserved	Reserved
54	Reserved	Reserved

Table 2-9. DSP Event Assignment (continued)

Event No.	Interrupt	Description
55	Reserved	Reserved
56	Reserved	Reserved
57	Reserved	Reserved
58	DSS_MSS_SW0	DSS_MSS_SW interrupt
59	DSS_MSS_SW1	DSS_MSS_SW interrupt
60	DSS_DMMSWINT5	Interrupt from DSS_DMM configurable
61	DSS_DMMSWINT6	Interrupt from DSS_DMM configurable
62	DSS_BSS_SW1	Radar SS SW Interrupt 0
63	DSS_BSS_SW2	Radar SS SW Interrupt 1
64	DSS_TPTC2_IRQ_DONE	DSS_TPTC2 (EDMA TPTC2) completion interrupt
65	DSS_TPTC2_IRQ_ERR	DSS_TPTC2 (EDMA TPTC2) Error Interrupt
66	DSS_TPTC3_IRQ_DONE	DSS_TPTC3 (EDMA TPTC3) completion interrupt
67	DSS_TPTC3_IRQ_ERR	DSS_TPTC3 (EDMA TPTC3) Error Interrupt
68	DSS_TPCC1_IRQ_DONE	DSS_TPCC1 (EDMA TPCC1) Global completion Interrupt
69	DSS_TPCC1_IRQ_ERR	DSS_TPCC1 (EDMA TPCC1) Error Interrupt
70	DSS_ADC_DATA_VALID_FALL/DSS_DMMS WINT3/DSS_DMMSWINT44	DSS_ADC Ping/Pong interrupt
71	DSS_UART_REQ0	DSS_SCI (UART) Req 0
72	DSS_UART_REQ1	DSS_SCI (UART) Req 1
73	DSS_RTI0_OVERFLOW_0	DSS_RTI Overflow 0
74	DSS_RTI0_OVERFLOW_1	DSS_RTI Overflow 1
75	DSS_RTI0_0	DSS_RTI Interrupt 0
76	DSS_RTI0_1	DSS_RTI Interrupt 1
77	DSS_RTI0_2	DSS_RTI Interrupt 2
78	DSS_RTI0_3	DSS_RTI Interrupt 3
79	DSS_RTI1_OVERFLOW_0	DSS_RTI2 Overflow 0
80	DSS_RTI1_OVERFLOW_1	DSS_RTI2 Overflow 1
81	DSS_RTI1_0	DSS_RTI2 Interrupt 0
82	DSS_RTI1_1	DSS_RTI2 Interrupt 1
83	DSS_RTI1_2	DSS_RTI2 Interrupt 2
84	DSS_RTI1_3	DSS_RTI2 Interrupt 3
85	DSS_BSS_MAILBOX_FULL	Interrupt indicating there is a message from MSS in the Mailbox BSS-DSS
86	DSS_BSS_MAILBOX_EMPTY	Interrupt indicating the MSS has read/ack the message DSP posted in the Mailbox DSS-BSS
87	GPIO_0_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
88	GPIO_1_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
89	GPIO_2_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
90	Reserved	Reserved
91	DSS_MSS_MAILBOX_FULL	Interrupt indicating there is a message from MSS in the Mailbox MSS-DSS
92	DSS_MSS_MAILBOX_EMPTY	Interrupt indicating the MSS has read/ack the message DSP posted in the Mailbox DSS-MSS
93	DSS_LOGICAL_FRAME_START/DSS_DMM SWINT1/DSS_DMMSWINT40	Logical Frame start interrupt
94	DSS_DMMSWINT7	Interrupt from DSS_DMM configurable
95	Reserved	Reserved
96	INTERR	DSP dropped CPU interrupt event
97	IDMA_ERR	Invalid IDMA parameters

Table 2-9. DSP Event Assignment (continued)

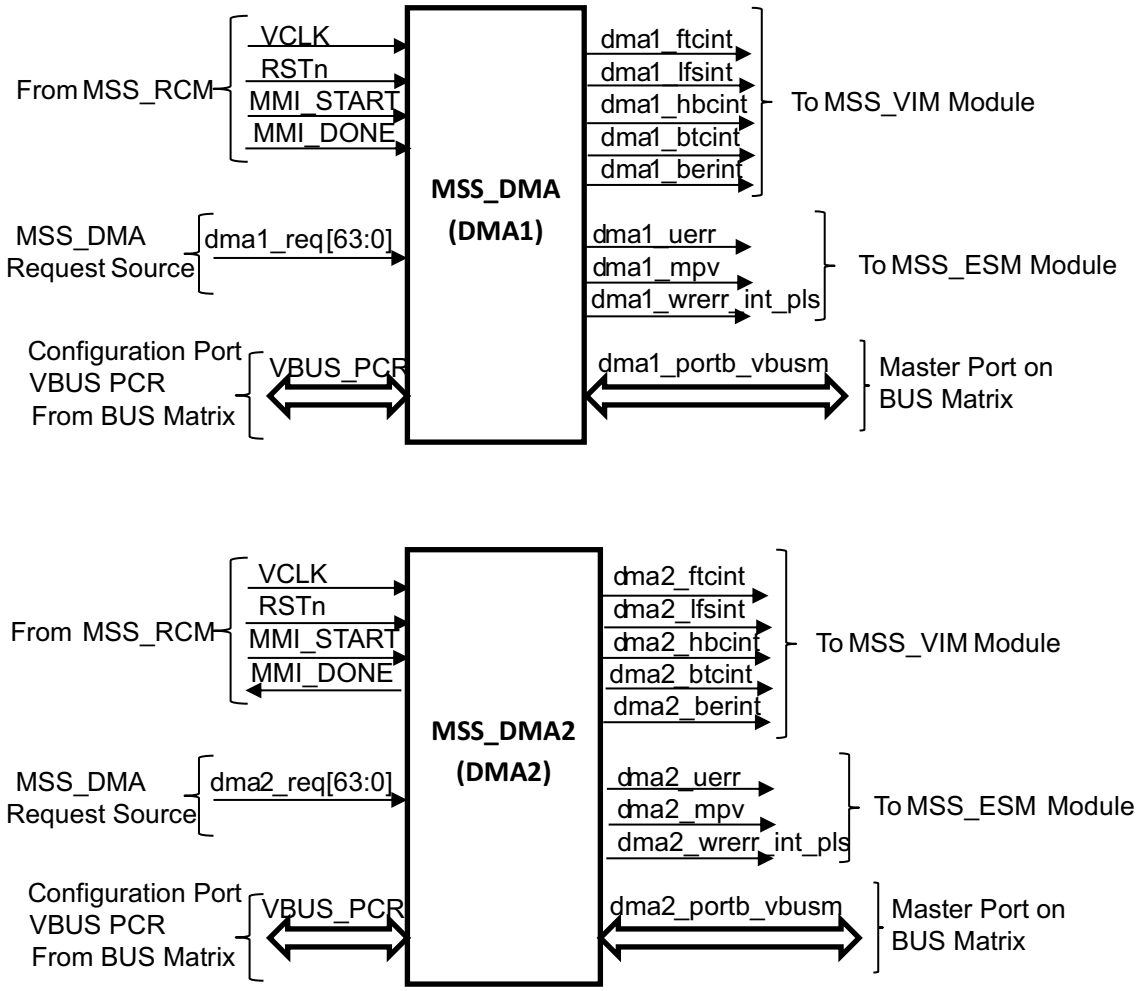
Event No.	Interrupt	Description
98	Reserved	Reserved
99	Reserved	Reserved
100	Reserved	Reserved
101	Reserved	Reserved
102	Reserved	Reserved
103	Reserved	Reserved
104	Reserved	Reserved
105	Reserved	Reserved
106	Reserved	Reserved
107	Reserved	Reserved
108	Reserved	Reserved
109	Reserved	Reserved
110	Reserved	Reserved
111	Reserved	Reserved
112	Reserved	Reserved
113	DSP_PMC_ED	DSS_DSP_L1P parity error
114	Reserved	Reserved
115	Reserved	Reserved
116	DSP_UMC_ED1	DSS_DSP_L2 ECC single error correction
117	DSP_UMC_ED2	DSS_DSP_L2 ECC double error detection
118	DSP_PDC_INT	Power down sleep interrupt
119	DSP_SYS_CMPA	CPU memory protection fault
120	DSP_L1P_CMPA	DSS_DSP_L1P CPU memory protection fault
121	DSP_L1P_DMPA	DSS_DSP_L1P DMA memory protection fault
122	DSP_L1D_CMPA	DSS_DSP_L1D CPU memory protection fault
123	DSP_L1D_DMPA	DSS_DSP_L1D DMA memory protection fault
124	DSP_L2_CMPA	DSS_DSP_L2 CPU memory protection fault
125	DSP_L2_DMPA	DSS_DSP_L2 DMA memory protection fault
126	DSP_EMC_CMPA	From EMC, CPU memory protection fault
127	DSP_EMC_BUSSERR	From EMC, bus error interrupt

2.3.4 Direct Memory Access Controller (MSS_DMA)

2.3.4.1 MSS_DMA Integration Diagrams

The device has two instances of DMA module, MSS_DMA and MSS_DMA2. Integration of the two DMA blocks in the device are shown in Figure 3-5 and .

Figure 2-6. Integration of MSS_DMA and MSS_DMA2 Module



2.3.4.2 MSS_DMA Features

- 64-bit OCP protocol to perform bus master accesses
- INCR-4 64-bit burst accesses
- Multithreading architecture allowing data of two different channel transfers to be interleaved during non-burst accesses
- 2-port configuration for parallel bus master
- Channels can be assigned to either high-priority queue or low-priority queue. Within each queue, fixed or round-robin priorities can be serviced
- Built-in ECC generation and evaluation logic for internal RAM-storing channel transfer information
- Supports multiple interrupt outputs for mapping to multiple interrupt controllers in multicore systems
- 48 requests can be mapped to any 32 channels
- Supports LE endianness
- External ECC Gen/Eval block of MSS_DMA support ECC generation for data transactions, and parity for address, and control signals (following Cortex-R5F standard)
- 8 MPU regions
- Channel-chaining capability
- Hardware and software MSS_DMA requests

- 8-, 16-, 32-, or 64-bit transactions supported
- Multiple addressing modes for source and destination (fixed, increment, offset)
- Auto-initiation

2.3.4.3 MSS_DMA Request Map

Both instances of MSS_DMA have 64 lines of request and are connected to identical input triggers, as shown in Table 3-10. This allows the two DMAs to trigger different types of transfers for the same request.

Table 2-10. MSS_DMA Request Map

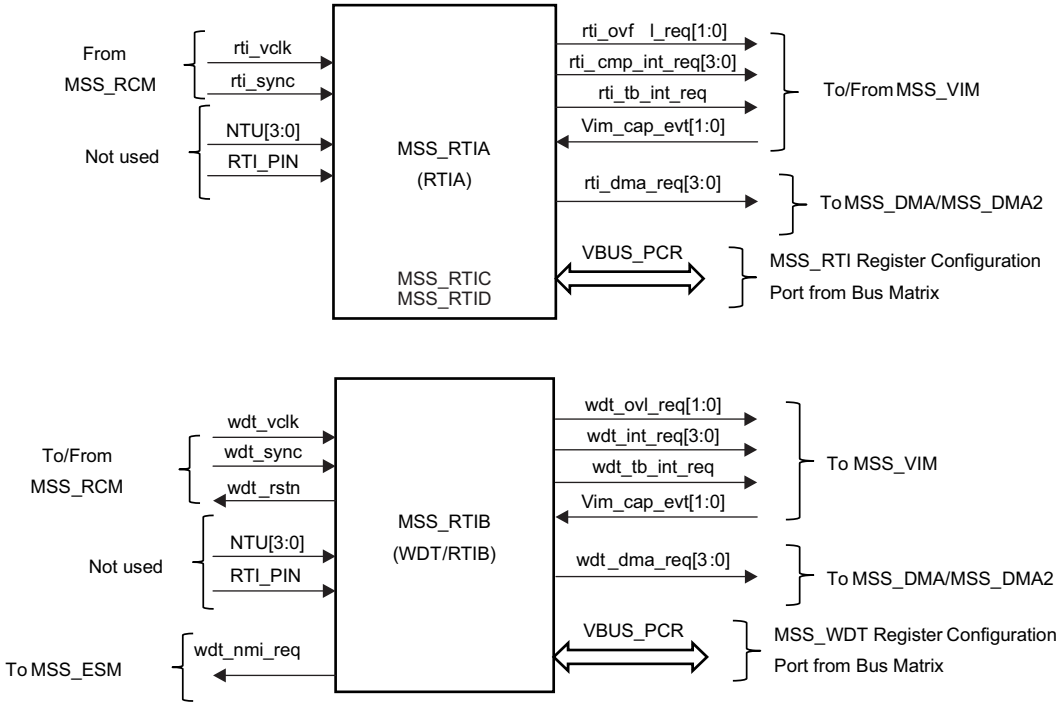
Module	DMA Request Sources	DMA Request
MSS_MIBSPIA	MSS_MIBSPIA Channel-1	DMAREQ[0]
MSS_MIBSPIA	MSS_MIBSPIA Channel-0	DMAREQ[1]
MSS_MIBSPIB	MSS_MIBSPIB	DMAREQ[2]
MSS_MIBSPIB	MSS_MIBSPIB	DMAREQ[3]
MSS_QSPI	MSS_QSPI DMA request	DMAREQ[4]
MSS_MIBSPIA	MSS_MIBSPIA Channel-3	DMAREQ[5]
MSS_DCAN	MSS_DCAN IF2	DMAREQ[6]
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) DMAREQ	DMAREQ[7]
MSS_DCAN	MSS_DCAN IF1	DMAREQ[8]
MSS_MIBSPIA	MSS_MIBSPIA Channel-5	DMAREQ[9]
MSS_I2C	MSS_I2C receive	DMAREQ[10]
MSS_I2C	MSS_I2C transmit	DMAREQ[11]
MSS_RTIA	MSS_RTIA DMAREQ0	DMAREQ[12]
MSS_RTIA	MSS_RTIA DMAREQ1	DMAREQ[13]
Reserved	Reserved	DMAREQ[14]
Reserved	Reserved	DMAREQ[15]
MSS_DCAN	MSS_DCAN IF3	DMAREQ[16]
MSS_MIBSPIA	MSS_MIBSPIA Channel-2	DMAREQ[17]
MSS_RTIA	MSS_RTIA DMAREQ2	DMAREQ[18]
MSS_RTIA	MSS_RTIA DMAREQ3	DMAREQ[19]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ0	DMAREQ[20]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ1	DMAREQ[21]
MSS_MIBSPIA	MSS_MIBSPIA Channel-4	DMAREQ[22]
MSS_ETPWM3A	MSS_ETPWM3A DMAREQ	DMAREQ[23]
MSS_RTIB (WDT/RTIB)	WDT/RTIB DMAREQ2	DMAREQ[24]
MSS_RTIB (WDT/RTIB)	WDT/RTIB DMAREQ3	DMAREQ[25]
MSS_MCRC (CRC)	MSS_MCRC (CRC) DMAREQ0	DMAREQ[26]
MSS_MCRC (CRC)	MSS_MCRC (CRC) DMAREQ1	DMAREQ[27]
MSS_SCIB (UART2)	MSS_SCIB (UART2) receive	DMAREQ[28]
MSS_SCIB (UART2)	MSS_SCIB (UART2) transmit	DMAREQ[29]
MSS_SCIA (UART1)	MSS_SCIA (UART1) receive	DMAREQ[30]
MSS_SCIA (UART1)	MSS_SCIA (UART1) transmit	DMAREQ[31]
MSS_GIO	MSS_GIO-0	DMAREQ[32]
MSS_GIO	MSS_GIO-1	DMAREQ[33]
MSS_GIO	MSS_GIO-2	DMAREQ[34]
MSS_ETPWM1A	MSS_ETPWM1A DMAREQ	DMAREQ[35]
Reserved	Reserved	DMAREQ[36]
MSS_MIBSPIB	MSS_MIBSPIB Channel-2	DMAREQ[37]
MSS_MIBSPIB	MSS_MIBSPIB Channel-3	DMAREQ[38]
MSS_ETPWM1B	MSS_ETPWM1B DMAREQ	DMAREQ[39]

Table 2-10. MSS_DMA Request Map (continued)

Module	DMA Request Sources	DMA Request
MSS_ETPWM2A	MSS_ETPWM2A DMAREQ	DMAREQ[40]
MSS_ETPWM2B	MSS_ETPWM2B DMAREQ	DMAREQ[41]
MSS_MIBSPIB	MSS_MIBSPIB Channel-4	DMAREQ[42]
MSS_MIBSPIB	MSS_MIBSPIB Channel-5	DMAREQ[43]
Reserved	Reserved	DMAREQ[44]
MSS_ETPWM3B	MSS_ETPWM3B DMAREQ	DMAREQ[45]
MSS_GIO	MSS_GIO-14	DMAREQ[46]
MSS_GIO	MSS_GIO-15	DMAREQ[47]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-0	DMAREQ[48]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-1	DMAREQ[49]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-2	DMAREQ[50]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-3	DMAREQ[51]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-4	DMAREQ[52]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-5	DMAREQ[53]
MSS_DTHE (Crypto/AES)	AES DMAREQ-0	DMAREQ[54]
MSS_DTHE (Crypto/AES)	AES DMAREQ-1	DMAREQ[55]
MSS_DTHE (Crypto/AES)	AES DMAREQ-2	DMAREQ[56]
MSS_DTHE (Crypto/AES)	AES DMAREQ-3	DMAREQ[57]
MSS_DTHE (Crypto/AES)	AES DMAREQ-4	DMAREQ[58]
MSS_DTHE (Crypto/AES)	AES DMAREQ-5	DMAREQ[59]
MSS_DTHE (Crypto/AES)	AES DMAREQ-6	DMAREQ[60]
MSS_DTHE (Crypto/AES)	AES DMAREQ-7	DMAREQ[61]
Reserved	Reserved	DMAREQ[62]
Reserved	Reserved	DMAREQ[63]

2.3.5 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB)

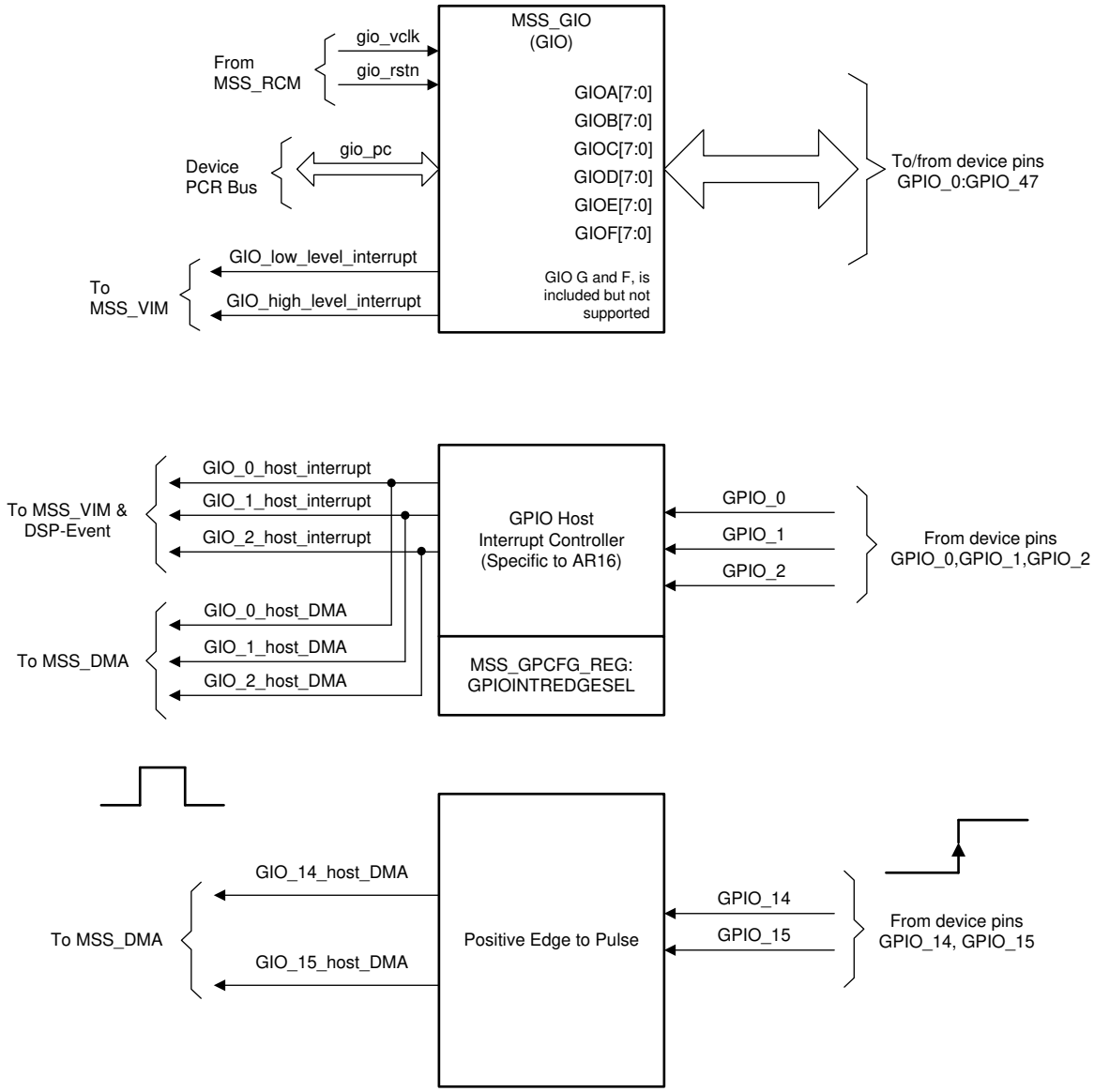
Figure 2-7. Integration of MSS_RTIA and MSS_RTIB, WDT Using the MSS_RTIB Module



2.3.6 General Purpose I/0 (MSS_GIO)

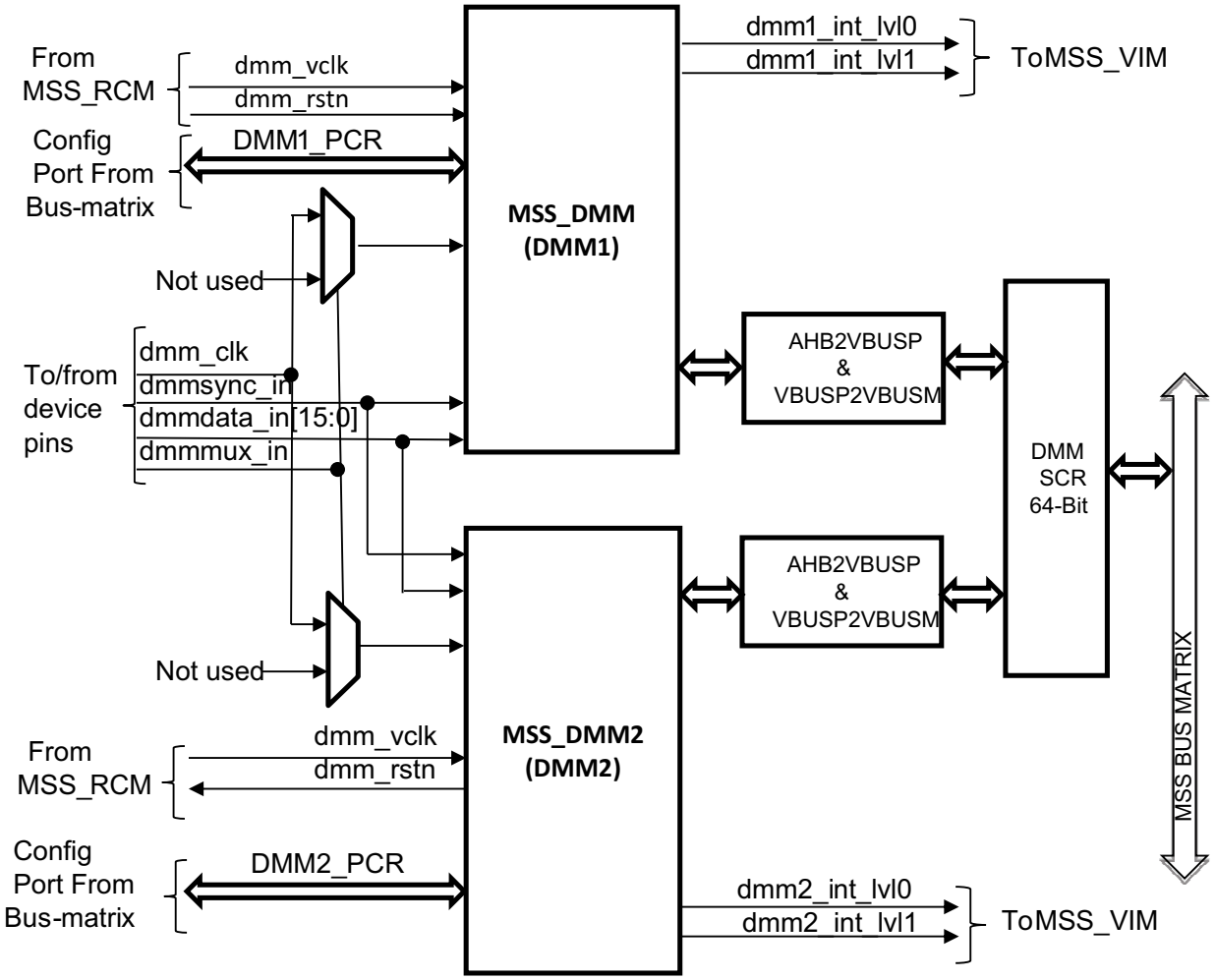
NOTE: Emulation mode and power-down mode (low-power mode) are not supported in the 16xx device.

Figure 2-8. Integration Block Diagram for MSS_GIO



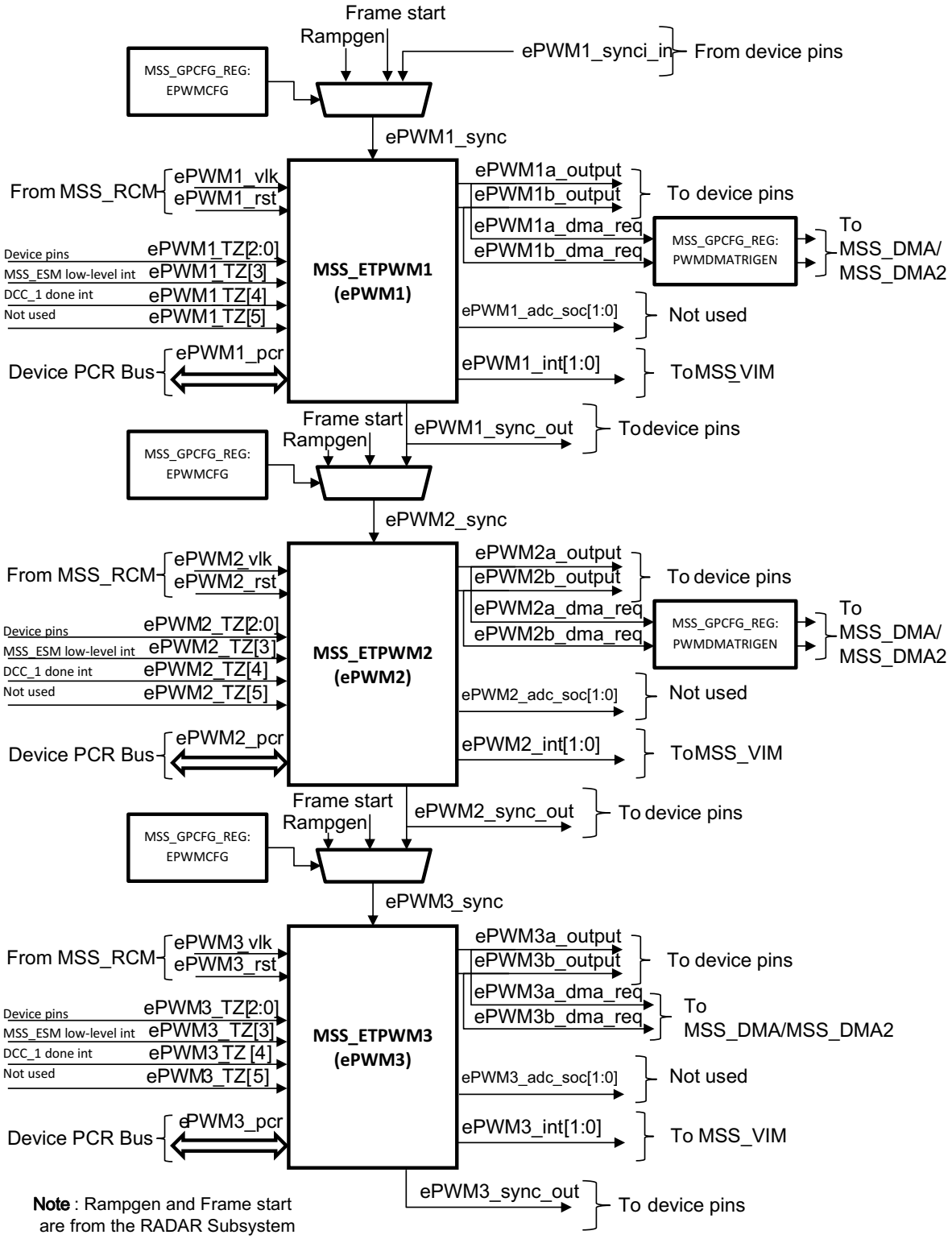
2.3.7 Data Modification Module (MSS_DMM)

Figure 2-9. MSS_DMM Integration



2.3.8 Enhanced Pulse Width Modulator (MSS_ETPWM)

Figure 2-10. Multiple MSS_ETPWM Modules



2.3.9 Vectored Interrupt Manager (MSS_VIM)

2.3.9.1 Interrupt Request Assignments

Table 2-11. Interrupt Request Assignments

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
MSS_ESM	MSS_ESM high-level interrupt(NMI)	0
Reserved	Reserved	1
MSS_RTIA	MSS_RTIA compare interrupt 0	2
MSS_RTIA	MSS_RTIA compare interrupt 1	3
MSS_RTIA	MSS_RTIA compare interrupt 2	4
MSS_RTIA	MSS_RTIA compare interrupt 3	5
MSS_RTIA	MSS_RTIA overflow interrupt 0	6
MSS_RTIA	MSS_RTIA overflow interrupt 1	7
MSS_RTIA	MSS_RTIA time-base	8
MSS_GIO	MSS_GIO high-level interrupt	9
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 0	10
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt1	11
MSS_MIBSPIA	MSS_MIBSPIA level 0 interrupt	12
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 2	13
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 3	14
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) overflow interrupt 0	15
MSS_DCAN	MSS_DCAN level 0 interrupt	16
MSS_MIBSPIB	MSS_MIBSPIB level 0 Interrupt	17
MSS_GIO host interrupt module	MSS_GIO GPIO_0_host_interrupt	18
MSS_MCRC (CRC)	MSS_MCRC (CRC) interrupt	19
MSS_ESM	MSS_ESM low-level interrupt	20
SYSTEM	Software-triggered interrupt 4	21
MSS Cortex R4F	MSS Cortex R4F interrupt PMU	22
MSS_GIO	MSS_GIO low-level interrupt	23
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) overflow interrupt 1	24
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) TB base interrupt	25
MSS_MIBSPIA	MSS_MIBSPIA level 0 interrupt	26
MSS_QSPI	MSS_QSPI interrupt	27
MSS_DMM	MSS_DMM S/W interrupt 38	28
MSS_DCAN	MSS_DCAN level 1 interrupt	29
MSS_MIBSPIB	MSS_MIBSPIB level 1 interrupt	30
MSS_DTHE (Crypto/SHA)	MSS_DTHE (Crypto/SHA) SHA -S interrupt	31
MSS_GIO host interrupt module	MSS_GIO GPIO_1_host_interrupt	32
MSS_DMA	MSS_DMA frame transfer complete interrupt	33
MSS_DMA	MSS_DMA last frame transfer start interrupt	34
Reserved	Reserved	35
MSS_DMM	MSS_DMM level -0 interrupt	36
MSS_DTHE (Crypto/SHA)	MSS_DTHE (Crypto/SHA) SHA -P interrupt	37
MSS_DTHE (Crypto/TRNG)	MSS_DTHE (Crypto/TRNG) TRNG interrupt	38

Table 2-11. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
MSS_DMA	MSS_DMA half-block transfer complete interrupt	39
MSS_DMA	MSS_DMA block transfer complete interrupt	40
MSS_DMA2	MSS_DMA2 frame block transfer complete interrupt	41
Reserved	Reserved	42
MSS_DMM	MSS_DMM level -1 interrupt	43
Reserved	Reserved	44
MSS_DMA2	MSS_DMA2 last frame complete interrupt	45
Reserved	Reserved	46
FPU	Floating point unit interrupt	47
MSS_GIO (GPIO host interrupt module)	MSS_GIO GPIO_2_host_interrupt	48
MSS_DMA2	MSS_DMA2 half-block transfer complete interrupt	49
MSS_DMA2	MSS_DMA2 block transfer complete interrupt	50
MSS_DMA2	MSS_DMA2 bus error interrupt	51
System	DSS to MSS software-triggered by register DSS_REG2:MSSSWIRQ:MSSSWIRQ1	52
MSS_DTHE (Crypto/PKA)	MSS_DTHE (Crypto/PKA) PKA module interrupt	53
MSS_DTHE (Crypto/AES)	MSS_DTHE (Crypto/AES) AES-S module interrupt	54
Reserved	Reserved	55
MSS_DTHE (Crypto/AES)	MSS_DTHE (Crypto/AES) AES-P module interrupt	56
MSS_DMM2	MSS_DMM2 level -0 interrupt	57
MSS_DMM2	MSS_DMM2 level -1 interrupt	58
Mailbox	DSS to MSS mailbox full interrupt	59
Mailbox	DSS to MSS mailbox empty interrupt	60
System	DSS to MSS software-triggered by register DSS_REG2:MSSSWIRQ:MSSSWIRQ2	61
MSS_DEBUGSS (Debug subsystem)	MSS_DEBUGSS (Debug subsystem) interrupt	62
DSPSS-MSS_STC	GEM MSS_STC done interrupt	63
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 0 interrupt	64
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 0 interrupt	65
MSS_I2C	MSS_I2C interrupt	66
MSS_DMM	MSS_DMM interrupt 34	67
MSS_DMM	MSS_DMM interrupt 35	68
MSS_DMM	MSS_DMM interrupt 36	69
MSS_DMA	MSS_DMA bus error interrupt	70
MSS_DMM/Radar subsystem	MSS_DMM interrupt 30 or Radar subsystem logical Frame Start	71
Reserved	Reserved	72
MSS_DMM	MSS_DMM interrupt 33	73
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 1 interrupt	74
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 1 interrupt	75
SYSTEM	Software-triggered interrupt 0	76

Table 2-11. Interrupt Request Assignments (continued)

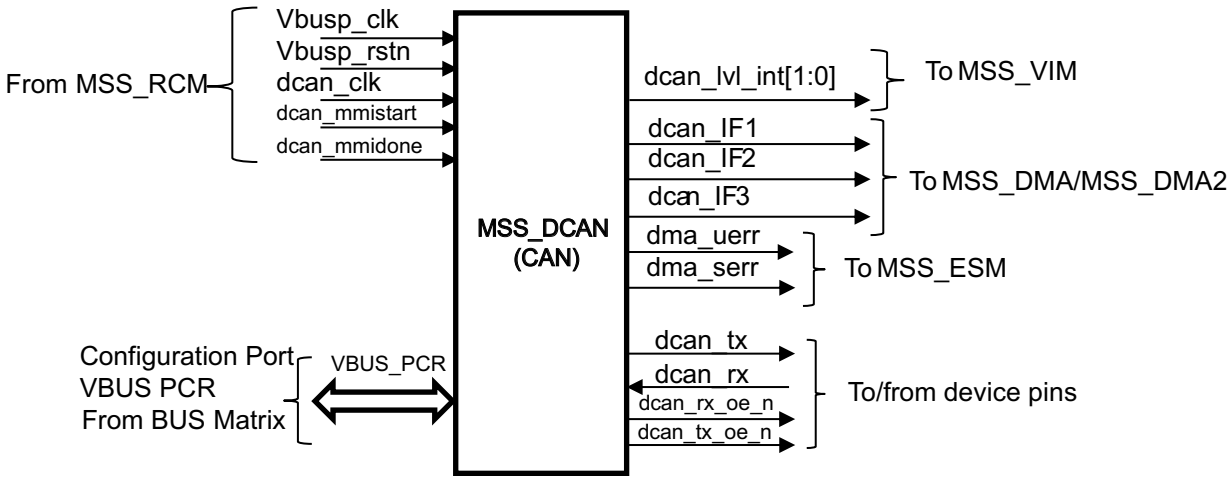
Module	VIM Interrupt Sources	Default VIM Interrupt Channel
SYSTEM	Software-triggered interrupt 1	77
SYSTEM	Software-triggered interrupt 2	78
SYSTEM	Software-triggered interrupt 3	79
Reserved	Reserved	80
Reserved	Reserved	81
MSS_DCCA	MSS_DCCA (dual clock compare) module1-done interrupt	82
MSS_DCCB	MSS_DCCB (dual clock compare) module2-done interrupt	83
SYSTEM	Software-triggered interrupt 5	84
MSS_PBIST	MSS_PBIST interrupt	85
MSS_DMM/DSS	GEM IRQ-7/MSS_DMM interrupt 32	86
Reserved	Reserved	87
Reserved	Reserved	88
Reserved	Reserved	89
Reserved	Reserved	90
Reserved	Reserved	91
Reserved	Reserved	92
Reserved	Reserved	93
Reserved	Reserved	94
MAILBOX	RADARSS to MSS mailbox interrupt	95
MAILBOX	RADARSS mailbox read complete interrupt sent from RADARSS to MSS	96
RADARSS	ADC valid fall interrupt	97
MSS_DMM/RADARSS	MSS_DMM interrupt 29/frame start interrupt/	98
RADARSS	Chirp start interrupt	99
RADARSS	Chirp end Interrupt	100
RADARSS	Frame end Interrupt	101
Reserved	Reserved	102
Reserved	Reserved	103
MSS_ETPWM1	ePWM1 interrupt-1	104
RADARSS-MSS_STC	MSS_STC done Interrupt	105
RadarSS	All RadarSS interrupts combined	106
MSS_ETPWM1	ePWM1 interrupt-2	107
MSS_ETPWM2	ePWM2 interrupt-1	108
MSS_ETPWM2	ePWM2 interrupt-2	109
MSS_ETPWM3	ePWM3 interrupt-1	110
MSS_ETPWM3	ePWM3 interrupt-2	111
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) interrupt	112
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) error interrupt	113
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) interrupt	114
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) error interrupt	115
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) interrupt	116
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) error interrupt	117
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) interrupt	118

Table 2-11. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
Reserved	Reserved	119
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) error interrupt	120
MSS_DMM	MSS_DMM interrupt 37	121
Reserved	Reserved	122
DSS_ADCBUF/MSS_DMM	Chirp available interrupt/MSS_DMM interrupt 31	123
MSS_PBIST	MSS_PBIST: Gem MSS_STC done	124
Reserved	Reserved	125
Reserved	Reserved	126
Reserved	Reserved	127

2.3.10 Controller Area Network (MSS_DCAN)

Figure 2-11. Integration Block Diagram for MSS_DCAN Module



2.3.11 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPI)

Figure 2-12. MSS_MIBSPIA Integration

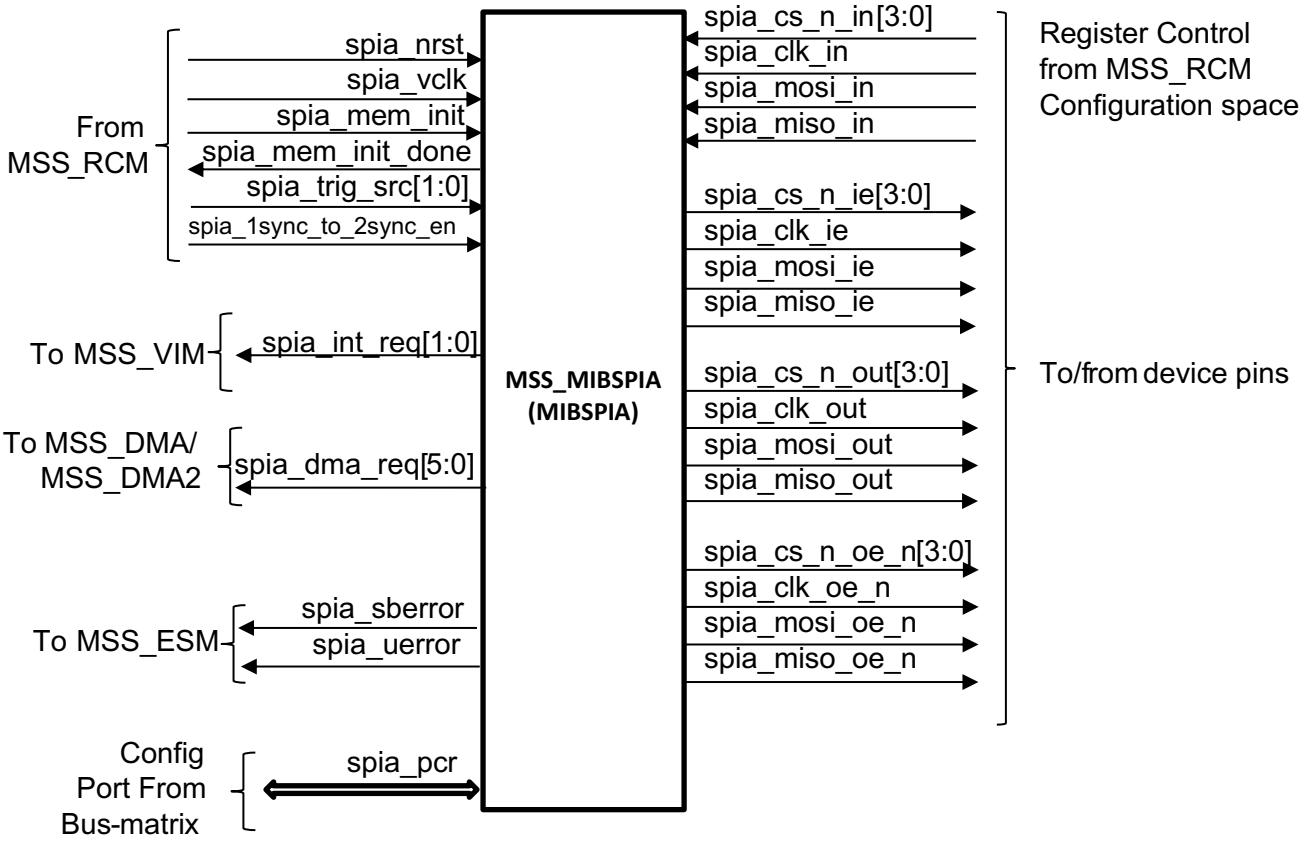
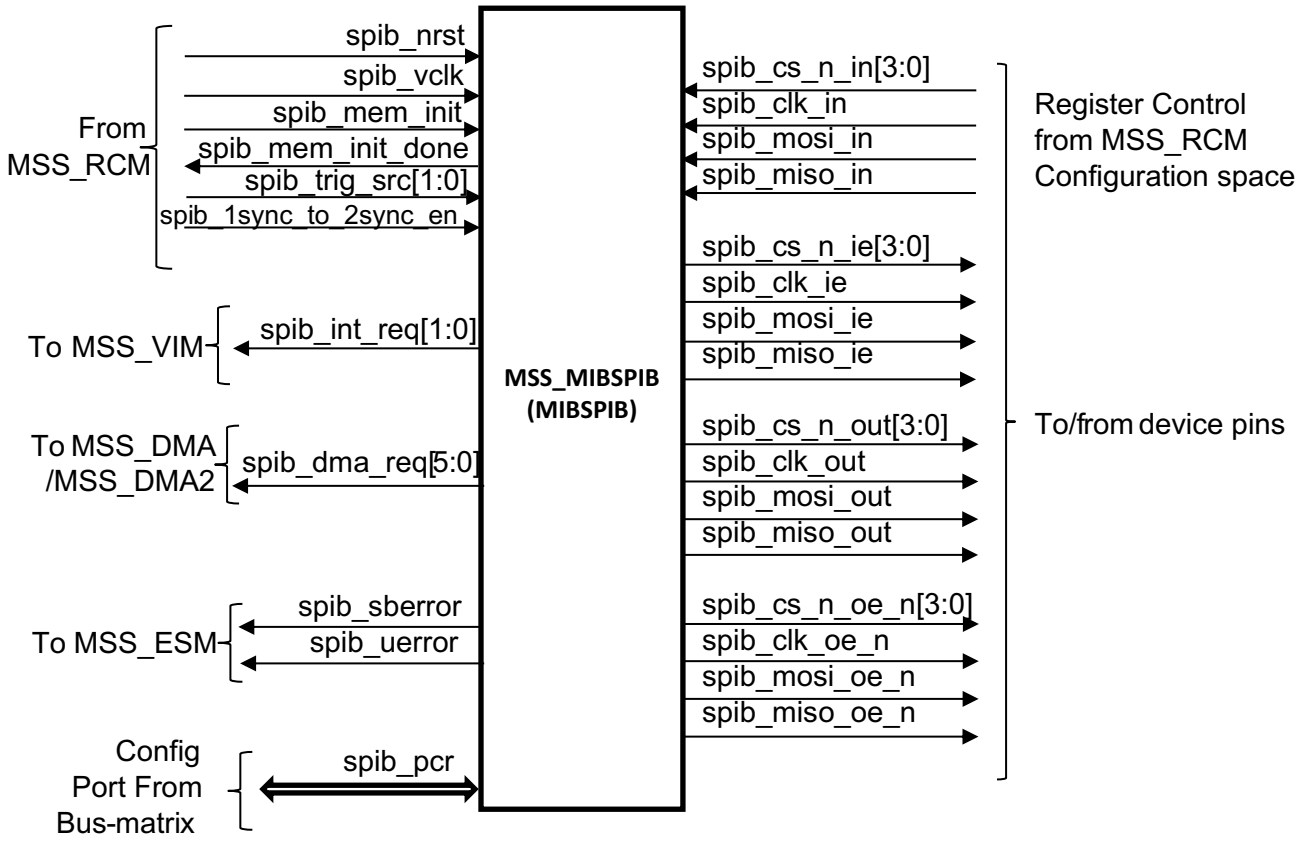


Figure 2-13. MSS_MIBSPIB Integration



2.3.12 Quad Serial Peripheral Interface (MSS_QSPI)

The MSS_QSPI module of the 16xx only supports one CS pin, (qspi1_cs).

2.3.13 Enhanced Direct Memory Access (EDMA)

2.3.13.1 EDMA Controller Integration

The 16xx device has two EDMA channel controllers (DSS_TPCC0 and DSS_TPCC1) on the device:

- DSS_TPCC0 (EDMA TPCC0) has two transfer controllers: DSS_TPTC0 (EDMA TPTC0) and DSS_TPTC1 (EDMA TPTC1)
- DSS_TPCC1 (EDMA TPCC1) has two transfer controllers: DSS_TPTC2 (EDMA TPTC2) and DSS_TPTC3 (EDMA TPTC3)

NOTE: The 16xx device does not support the region interrupt feature of the EDMA peripheral. Only the global interrupt feature of the EDMA module is supported.

Table 2-12. DSS_TPCC Configuration

	DSS_TPCC0 (EDMA TPCC0)	DSS_TPCC1 (EDMA TPCC1)
Number of MSS_DMA channels	64	64
Number of PaRAM entires	128	256
Number of QDMA channels	8	8

Table 2-12. DSS_TPCC Configuration (continued)

	DSS_TPCC0 (EDMA TPCC0)	DSS_TPCC1 (EDMA TPCC1)
Number of event queues	2	2
Memory protection existence	No	No
Channel mapping	No	No
Number of TCs (transfer controllers)	2	2

Table 2-13. DSS_TPTC Configuration

	DSS_TPTC[0-1]	DSS_TPTC[2-3]
FIFO size	512 bytes	128 bytes
TR pipe depth	2	2
Bus width	16 bytes	16 bytes

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 3-13 and Figure 3-14 show the EDMA controller integration.

Figure 2-14. EDMA Controller Integration (1 of 2)

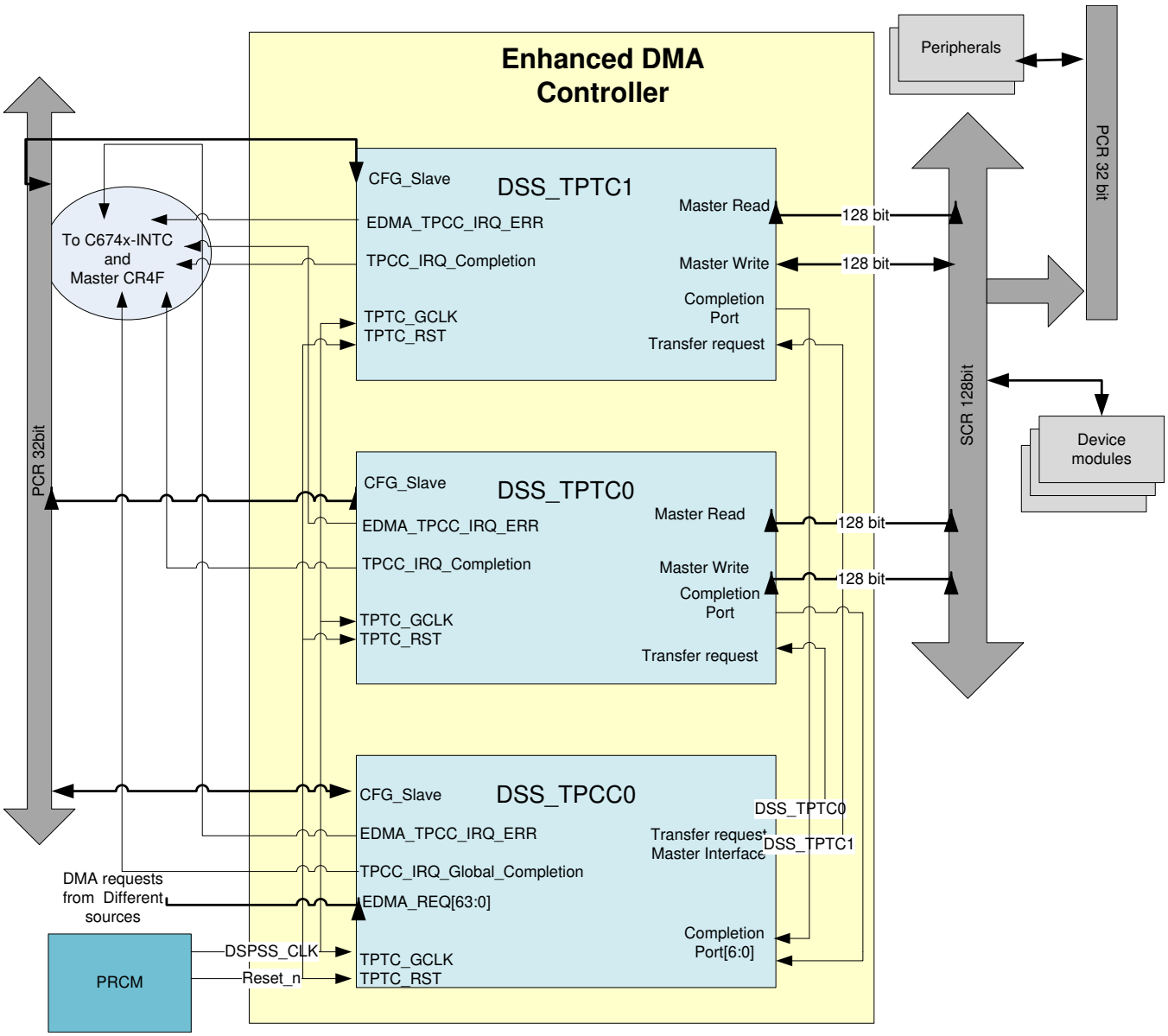
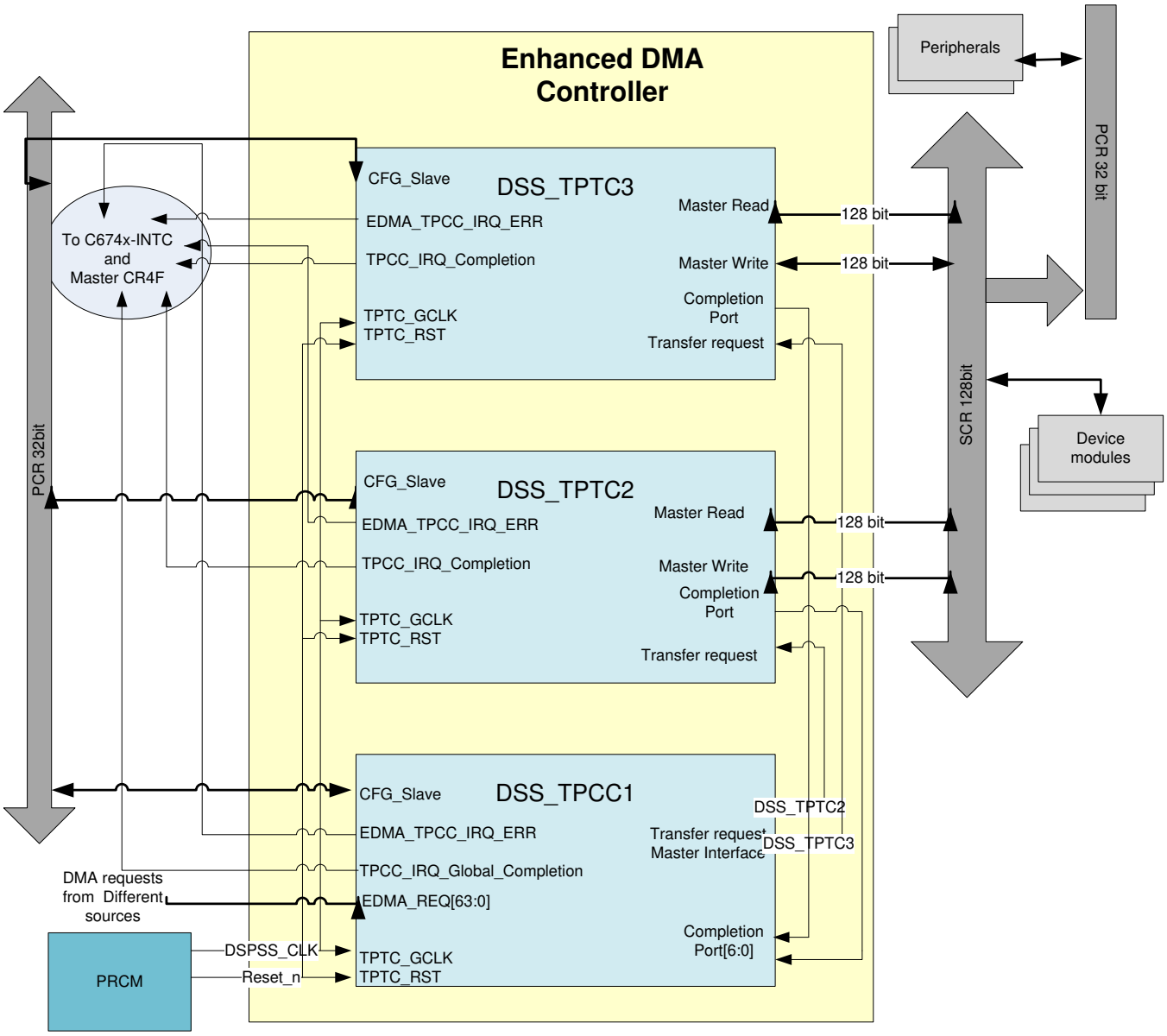


Figure 2-15. EDMA Controller Integration (2 of 2)



2.3.13.2 EDMA Request Map

Table 2-14. EDMA Request Map

Request Number	Hardware Event
DSS_TPCC0 (EDMA TPCC0) DMA	
0	DSS_CBUFF_DMA_REQ_0
1	DSS_CBUFF_DMA_REQ_1
2	DSS_CBUFF_DMA_REQ_2
3	DSS_CBUFF_DMA_REQ_3
4	DSS_CBUFF_DMA_REQ_4
5	DSS_CBUFF_DMA_REQ_5
6	DSS_CBUFF_DMA_REQ_6
7	RESERVED

Table 2-14. EDMA Request Map (continued)

Request Number	Hardware Event
8	Frame Start/DSS_DMMSWINT9/DSS_DMMSWINT39
9	Chirp Available/DSS_DMMSWINT11/DSS_DMMSWINT43
10	RESERVED
11	RESERVED
12	RESERVED
13	RESERVED
14	RESERVED
15	RESERVED
16	RESERVED
17	RESERVED
18	RESERVED
19	RESERVED
20	RESERVED
21	RESERVED
22	RESERVED
23	RESERVED
24	RESERVED
25	RESERVED
26	RESERVED
27	RESERVED
28	RESERVED
29	RESERVED
30	RESERVED
31	RESERVED
32	RESERVED
33	DSS_MCRC_DMA_REQ_0
34	DSS_MCRC_DMA_REQ_1
35	FRC_EVENT_GEN_0
36	FRC_EVENT_GEN_1
37	FRC_EVENT_GEN_2
38	FRC_EVENT_GEN_3
39	RESERVED
40	LOGICAL_FRAME_START/DSS_DMMSWINT10/DSS_DMMSWINT40
41	ADC_DATA_VALID_FALL/DSS_DMMSWINT12/DSS_DMMSWINT44
42	UART_DMA_REQ_0
43	UART_DMA_REQ_1
44	DMMSW_INT_13
45	DMMSW_INT_14
46	DMMSW_INT_15
47	DMMSW_INT_16
48	DMMSW_INT_17
49	GPIO_0_host_interrupt
50	GPIO_1_host_interrupt
51	GPIO_2_host_interrupt
52	RTI1_DMA_REQ_0
53	RTI1_DMA_REQ_1
54	RTI1_DMA_REQ_2

Table 2-14. EDMA Request Map (continued)

Request Number	Hardware Event
55	RTI1_DMA_REQ_3
56	RTI2_DMA_REQ_0
57	RTI2_DMA_REQ_1
58	RTI2_DMA_REQ_2
59	RTI2_DMA_REQ_3
60	RESERVED
61	RESERVED
62	RESERVED
63	DMMSW_INT_18
DSS_TPCC1 (EDMA TPCC1) MSS_DMA	
0	DSS_CBUFF_DMA_REQ_0
1	DSS_CBUFF_DMA_REQ_1
2	DSS_CBUFF_DMA_REQ_2
3	DSS_CBUFF_DMA_REQ_3
4	DSS_CBUFF_DMA_REQ_4
5	DSS_CBUFF_DMA_REQ_5
6	DSS_CBUFF_DMA_REQ_6
7	RESERVED
8	FRAME_START/DSS_DMMSWINT19/DSS_DMMSWINT39
9	CHIRP_AVIALABLE/DSS_DMMSWINT21/DSS_DMMSWINT43
10	RESERVED
11	RESERVED
12	RESERVED
13	RESERVED
14	RESERVED
15	RESERVED
16	RESERVED
17	RESERVED
18	RESERVED
19	RESERVED
20	RESERVED
21	RESERVED
22	RESERVED
23	RESERVED
24	RESERVED
25	RESERVED
26	RESERVED
27	RESERVED
28	RESERVED
29	RESERVED
30	RESERVED
31	RESERVED
32	RESERVED
18	RESERVED
19	RESERVED
20	RESERVED
21	RESERVED

Table 2-14. EDMA Request Map (continued)

Request Number	Hardware Event
22	RESERVED
23	RESERVED
24	RESERVED
25	RESERVED
26	RESERVED
27	RESERVED
28	RESERVED
29	RESERVED
30	RESERVED
31	RESERVED
32	RESERVED
33	DSS_MCRC_DMA_REQ_0
34	DSS_MCRC_DMA_REQ_1
35	FRC_EVENT_GEN_0
36	FRC_EVENT_GEN_1
37	FRC_EVENT_GEN_2
38	FRC_EVENT_GEN_3
39	RESERVED
40	LOGICAL_FRAME_START/DSS_DMMSWINT20/DSS_DMMSWINT40
41	ADC_DATA_VALID_FALL/DSS_DMMSWINT22/DSS_DMMSWINT44
42	UART_DMA_REQ_0
43	UART_DMA_REQ_1
44	DMMSW_INT_23
45	DMMSW_INT_24
46	DMMSW_INT_25
47	DMMSW_INT_26
48	DMMSW_INT_27
49	GPIO_0_host_interrupt
50	GPIO_1_host_interrupt
51	GPIO_2_host_interrupt
52	RTI1_DMA_REQ_0
53	RTI1_DMA_REQ_1
54	RTI1_DMA_REQ_2
55	RTI1_DMA_REQ_3
56	RTI2_DMA_REQ_0
57	RTI2_DMA_REQ_1
58	RTI2_DMA_REQ_2
59	RTI2_DMA_REQ_3
60	RESERVED
61	RESERVED
62	RESERVED
63	DMMSW_INT_28

2.3.14 Error Signaling Module (MSS_ESM/DSS_ESM)

The 16xx device has two instances of the Error Signaling Module (MSS_ESM/DSS_ESM), shown in Figure 3-15.

Figure 2-16. 16xx MSS_ESM/DSS_ESM Integration Diagram

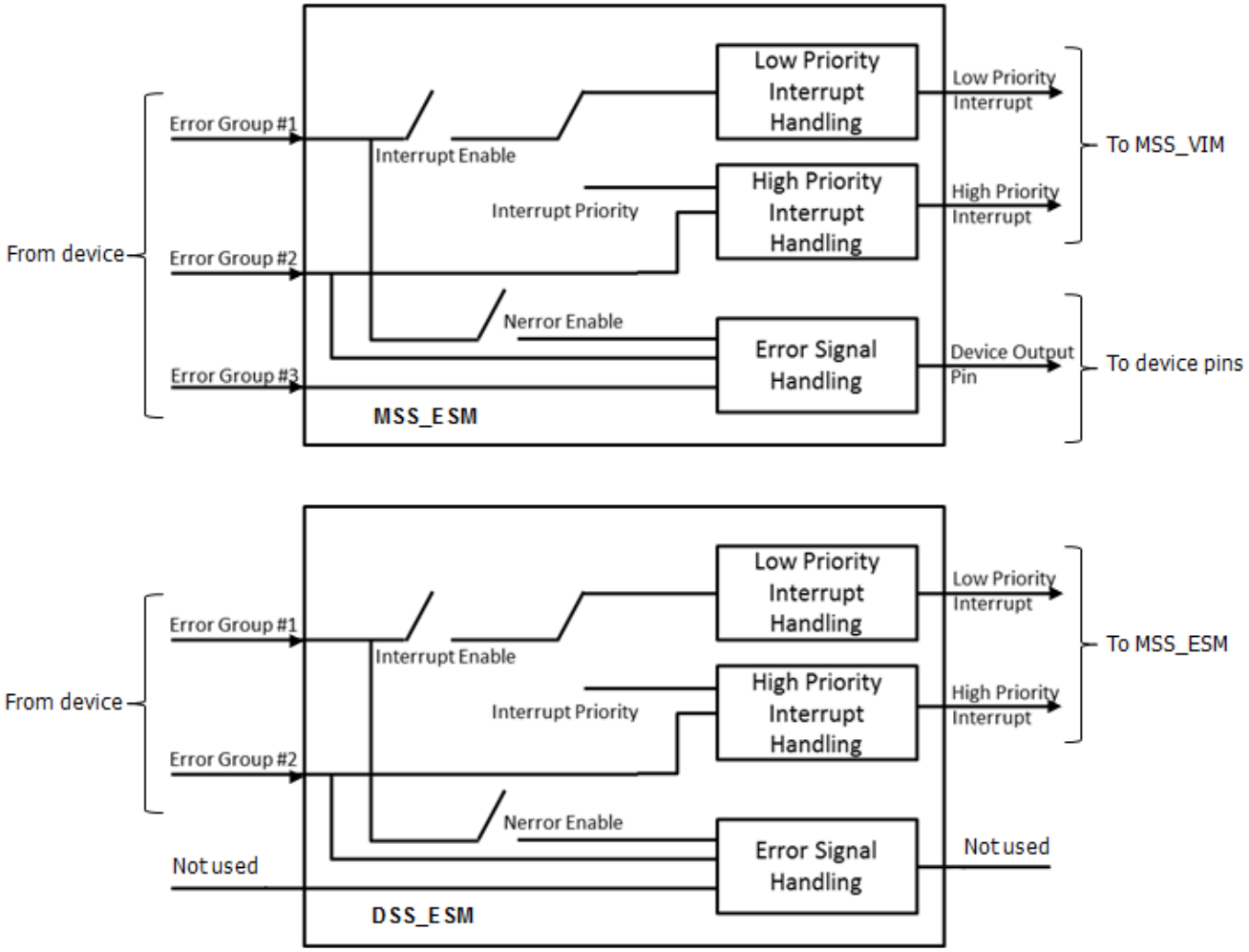


Table 3-15 shows the mapping on the input error inputs to the ESM module from various error sources available for hardware diagnostics within the device.

Table 2-15. MSS_ESM Mapping

MSS_ESM Group 1		Channel Type	Description
63	ANA_LIMP_MODE	Error Signal	Error signal at device boot-up, if the CLK monitor finds the REF CLK to be outside the permissible range of frequency
62	MSS_DCCB_ERR	Error Signal	MSS_DCCB frequency comparison error
61	MAILBOX_BSS2MSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2MSS
60	MAILBOX_BSS2MSS _REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_BSS2MSS
59	MAILBOX_MSS2BSS _FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2BSS
58	MAILBOX_MSS2BSS _REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2BSS
57	BSS_CRITICAL_ERR	Error Signal	BSS critical Error Indication. Mask control to be configured in IRQ_CTL for individual error signals.
56	Reserved	Reserved	Reserved

Table 2-15. MSS_ESM Mapping (continued)

MSS_ESM Group 1		Channel Type	Description
55	CLOCK_SUPPLY_ERR	Error Signal	Clock and Supply Errors from Analog. Mask control to be configured in IRQ_CTL for individual error signals
54	Reserved	Reserved	Reserved
53	MAILBOX_DSS2MSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2MSS
52	MAILBOX_DSS2MSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_DSS2MSS
51	Reserved	Reserved	Reserved
50	Reserved	Reserved	Reserved
49	MSS_MIBSPIB_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MIBSPIB multi-buffer (RXRAM/TXRAM)
48	MSS_MCRC_ERR	Error Signal	MSS_MCRC Comparison Error
47	Reserved	Reserved	Reserved
46	Reserved	Reserved	Reserved
45	MSS_MIBSPIB_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MIBSPIB multi-buffer (RXRAM/TXRAM)
44	Reserved	Reserved	Reserved
43	MAILBOX_MSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2DSS
42	MAILBOX_MSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2DSS
41	DSS_ESM_GP1_ERR	Error Signal	DSS_ESM Low priority Interrupt
40:39	Reserved	Reserved	Reserved
38	DSS_CBUFF_SAFETY_ERR	Error Signal	CHIRP ERROR or CRC ERROR from DSS_CBUFF
37	DSS_ESM_GP2_ERR	Error Signal	DSS_ESM High priority Interrupt
36	DSS_TPTC1_WR_MPU_ERR	Error Signal	DSS_TPTC1 write port MPU error
35	DSS_TPTC1_RD_MPU_ERR	Error Signal	DSS_TPTC1 read port MPU error
34	HVMODE_ERR	Error Signal	Error indication from IO Supply (Supply detector for dual-voltage IOs)
33	MSS_DCAN_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_DCAN Message RAM (FRAM/SRAM)
32	MSS_TCMA_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for Cortex R4F MSS_TCMA
31	Reserved	Reserved	Reserved
30	MSS_DCCA_ERR	Error Signal	MSS_DCCA frequency comparison error
29	DSS_TPTC0_WR_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
28	MSS_TCMB1_REPAIR_ERR	Alert Signal	Single-bit error/reserved indication for MSS_TCMB1
27	MSS_STC_ERR	Error Signal	MSS_STC Error indication for MSS Cortex R4F
26	MSS_TCMB0_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_TCMB0
25	MSS_MIBSPIA_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MIBSPIA multi-buffer (RXRAM/TXRAM)
24	MSS_DMA2_MEM_PARITY_ERR	Error Signal	Parity Error for MSS_DMA2 memory
23	MSS_DMA2_MPU_ERR	Error Signal	Error indication from MPU of MSS_DMA2
22	FRC_COMPARE_ERR	Error Signal	Lockstep comparison error from Free running Counter (FRC) in BSS
21	MSS_DCAN_RAM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_DCAN Message Memory (FRAM/SRAM)
20	MSS_VIM_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_VIM_RAM
19	Reserved	Reserved	Reserved
18	DSS_TPTC0_RD_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
17	MSS_MIBSPIA_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MIBSPIA (RXRAM/TXRAM)
16	MSS_SECURE_RAM_FATAL_ERR	Error Signal	Multi-bit uncorrectable error indication for MSS_DTHE/SECURE_RAM
15	MSS_VIM_RAM_FATAL_ERR	Error Signal	Multi-bit uncorrectable error indication for MSS_VIM_RAM

Table 2-15. MSS_ESM Mapping (continued)

MSS_ESM Group 1		Channel Type	Description
14	MSS_SECURE_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_DTHE/SECURE_RAM
13	Reserved	Reserved	Reserved
12	MAILBOX_BSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2DSS
11	MAILBOX_BSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_BSS2DSS
10	MAILBOX_DSS2BSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2BSS
9	DSS_CBUFF_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_CBUFF FIFO
8	DSS_CBUFF_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_CBUFF FIFO
7	DSS_TPCC_PARITY_ERR	Error Signal	Parity error from DSS_TPCC (EDMA Channel Controller)
6	NU	Reserved	Reserved
5	MAILBOX_DSS2BSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_DSS2BSS
4	MSS_CCCB_ERR	Error Signal	MSS_CCCB(Clock compare core) frequency comparison error
3	MSS_DMA_MEM_PARITY_ERR	Error Signal	Parity Error for DMA1 memory
2	MSS_DMA_MPU_ERR	Error Signal	Error indication from MPU of MSS _DMA
1	MSS_CCCA_ERR	Error Signal	MSS_CCCA(Clock compare core) frequency comparison error
0	NERROR_PAD_IN	Error Signal	Nerror from PAD looped in
MSS_ESM Group 2			
31:26	NU		
25	Reserved	Reserved	Reserved
24	MSS_RTIB_NMI	Error Signal	Watchdog Non-mask able interrupt
23:17	NU		
16	MSS_CR4F_LIVELOCK_ERR	Error Signal	Cortex R4F Live lock error
15:9	NU		
8	MSS_TCMB1_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB1
7	NU		
6	MSS_TCMB0_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB0
5	NU		
4	MSS_TCMA_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMA
3	NU		
2	Reserved	Reserved	Reserved
1:0	NU		
MSS_ESM Group 3			
31-8	NU		
7	MSS_TCMA_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMA
6	NU		
5	MSS_TCMB1_FATAL_ERR	Error Signal	Multi -bit error indication for MSS_TCMB1
4	NU		
3	MSS_TCMB0_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMB0
2	NU		
1	EFC_AUTOLOAD_ERR	Error Signal	Efuse Auto-load error
0	NU		

Table 2-16. DSS_ESM Mapping

DSS_ESM Group 1		Channel Type	Description
63:57	Reserved	Reserved	Reserved
56	DSS_DSP_L1P_PARITY_ERR	Error Signal	DSS_DSP_L1P Parity Error
55	MAILBOX_MSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2DSS
54	MAILBOX_DSS2MSS_REPAIR_ERR	Error Signal	Single-bit error/repair indication from MAILBOX_DSS2MSS
53	MAILBOX_DSS2MSS_FATAL_ERR	Alert Signal	Multi-bit error indication from MAILBOX_DSS2MSS
52	MAILBOX_MSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2DSS
51	MAILBOX_BSS2DSS_REPAIR_ERR	Alert Signal	Single-bit repair indication from MAILBOX_BSS2DSS
50	MAILBOX_DSS2BSS_REPAIR_ERR	Alert Signal	Single-bit repair indication from MAILBOX_DSS2BSS
49	MAILBOX_BSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2DSS
48	MAILBOX_DSS2BSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2BSS
47:32	Reserved	Reserved	Reserved
31	DSS_CFG_MSTID_MPU_ERR	Error Signal	Error from Master ID based MPU on the DSS_CFG_MSTID_MPU Configuration address space
30:29	Reserved	Reserved	Reserved
28	DSS_DATA_TXFR_RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_DATA_TXFR_RAM
27	DSS_DATA_TXFR_RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_DATA_TXFR_RAM
26	DSS_STC_ERR	Error Signal	Error from DSS_STC (Self-test Controller)
25	DSS_DSP_L2_UMAP_ECC_REPAIR_ERR	Alert Signal	Single bit repair indication for DSS_DSP_L2_UMAP0 or DSS_DSP_L2_UMAP1
24	DSS_HSRAM1_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_HSRAM1
23	DSS_HSRAM1_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_HSRAM1
22:19	Reserved	Reserved	Reserved
18	DSS_ADCBUF_PONG_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_ADCBUF Pong Memory
17	DSS_ADCBUF_PONG_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_ADCBUF Pong Memory
16	DSS_ADCBUF_PING_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_ADCBUF Ping Memory
15	DSS_ADCBUF_PING_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_ADCBUF Ping Memory
14	DSS_TPTC3_WR_MPU_ERR	Error Signal	DSS_TPTC3 write port MPU error
13	DSS_TPTC3_RD_MPU_ERR	Error Signal	DSS_TPTC3 read port MPU error
12	DSS_TPTC2_WR_MPU_ERR	Error Signal	DSS_TPTC2 write port MPU error
11	DSS_TPTC2_RD_MPU_ERR	Error Signal	DSS_TPTC2 read port MPU error
10	DSS_TPCC1_PARITY_ERR	Error Signal	Parity error from DSS_TPCC1 (EDMA Channel Controller)
9	DSS_CBUFF_SAFETY_ERR	Error Signal	CHIRP ERROR or CRC ERROR from DSS_CBUFF
8	DSS_TPTC1_WR_MPU_ERR	Error Signal	DSS_TPTC1 write port MPU error
7	DSS_TPTC1_RD_MPU_ERR	Error Signal	DSS_TPTC1 read port MPU error
6	DSS_TPTC0_WR_MPU_ERR	Error Signal	DSS_TPTC0 write port MPU error
5	DSS_TPTC0_RD_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
4	DSS_CBUFF_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_CBUFF FIFO
3	DSS_CBUFF_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_CBUFF FIFO
2	DSS_TPCC_PARITY_ERR	Error Signal	Parity error from DSS_TPCC (EDMA Channel Controller)
1	DSS_L3RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_L3RAM Memory
0	DSS_L3RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_L3RAM Memory
DSS_ESM Group 2			
31:6	Reserved	Reserved	Reserved
5	DSS_DSP_L2_UMAP_ECC_FATAL_ERR	Error Signal	Multi-bit error indication for DSS_DSP_L2_UMAP0 or DSS_DSP_L2_UMAP1
4	DSP_PBIST_ERR	Error Signal	DSP_PBIST (Memory Test) Fail Error

Table 2-16. DSS_ESM Mapping (continued)

DSS_ESM Group 1		Channel Type	Description
3	DSS_STC_ERR	Error Signal	Error from DSS_STC (Self-test Controller)
2	DSS_RTI_NMI	Error Signal	NMI from DSS_RTI (Watchdog)
1	DSS_DSP_L2_UMAP1_PARITY_ERR	Error Signal	Error from byte level parity comparison logic for DSS_DSP_L2_UMAP1
0	DSS_DSP_L2_UMAP0_PARITY_ERR	Error Signal	Error from byte level parity comparison logic for DSS_DSP_L2_UMAP0

2.3.15 High-Speed Interface (HSI)

Table 3-17 lists the high-speed interfaces available for the 16xx device.

Table 2-17. 16xx High-Speed Interfaces

CSI2	N/A
LVDS	2 lanes

2.3.16 Handshake RAM (DSS_HSRAM1)

The 16xx device has 32KB memory for HSRAM in the DSP subsystem.

2.4 18xx Introduction

2.4.1 18xx Overview

The 18xx is highly integrated single-chip RADAR device in TI's 45-nm low-power RFCMOS technology, a FCBGA 0.65-mm pitch automotive-grade package.

2.4.1.1 Features

- Frequency-Modulated Continuous Wave Radio Frequency Transceiver With 76-81-GHz Band
- Supports Three Transmitter Chains and Four Receiver Chains
- Chirp Profiles With Programmable Period and Slope
- 40-MHz, 50-MHz, 80-MHz, and 100-MHz XTAL/OSC Reference Input Clock
- 12, 14, and 16-bit Real/Complex ADC With Variable Baseband ADC Sampling Rates up to 18.75 MHz at 12-bits Complex
- Cortex R4F at 200-MHz Application Processor for Control Functionality and Safety-Critical Algorithms
- C674x DSP at up to 600 MHz for RADAR Data Processing
- Cortex R4F– Radio Processor at 200 MHz for Continuous Monitoring and Calibration of Analog/RF Functionality
- On-Chip Multicore Debug Support
- Customer-Programmable Efuse Support
- Up to 1024KB of L3 Shared Memory Support
- High-Performance Data Transfer With Multiple DMA and EDMA-TPCC Engines
- CAN Support for ECU Interface
- QSPI Serial Flash Support
- MIBSPI, SPI, I2C, and UART Serial Interfaces Support
- Hardware in Loop (HIL) Support
- Two-Lane Serial LVDS Interface Support
- AES, SHA, PKA, and TRNG Engines for Security

- 3 EPWM (Three Enhanced Pulse Width Modulator)
- Hardware Accelerator for FFT, Filtering, and CFAR Processing

2.4.2 18xx Description

2.4.2.1 Block Diagram

Figure 2-17 shows the block diagram of the 18xx device.

Figure 2-17. 18xx Block Diagram

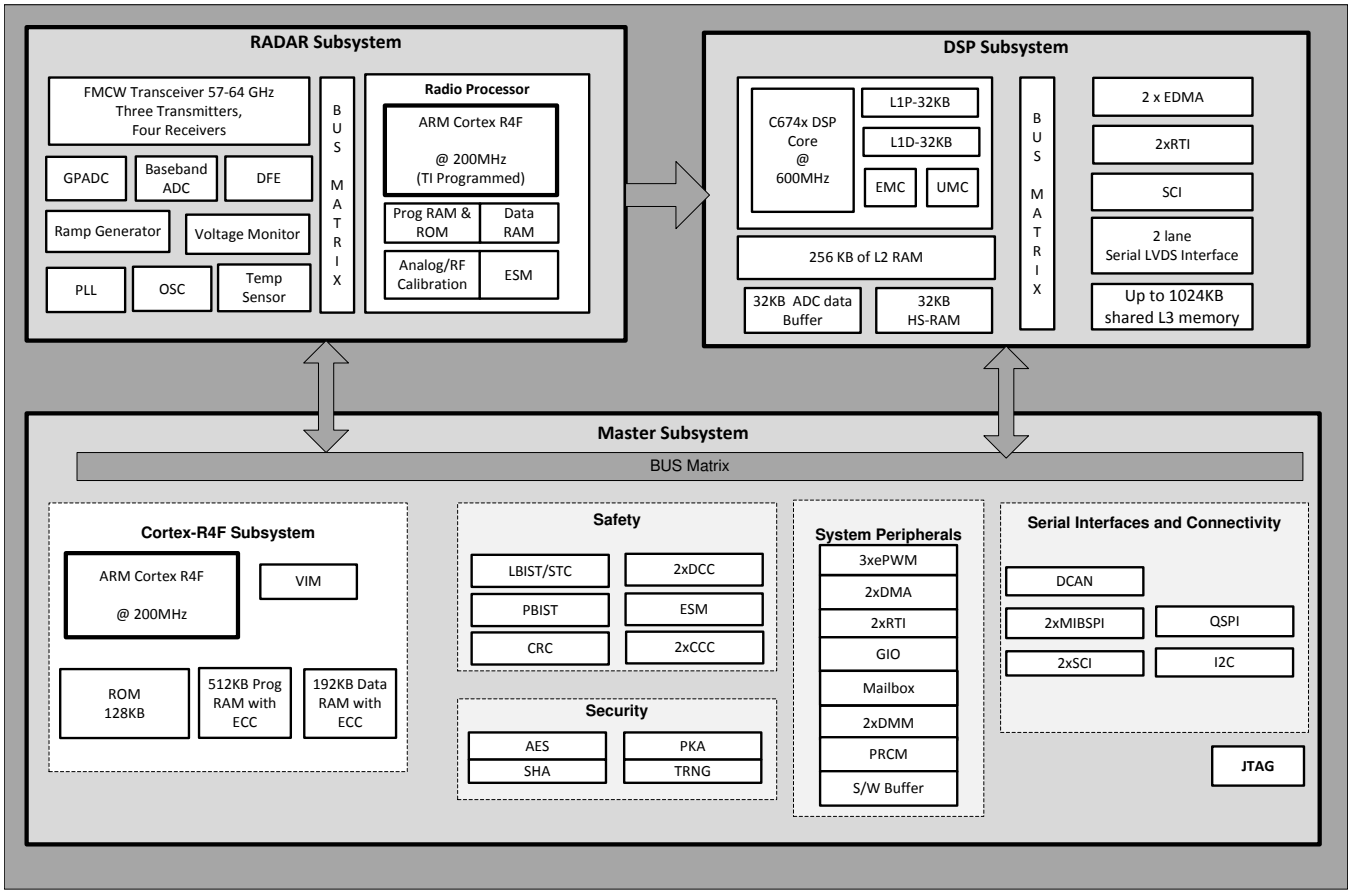


Table 2-18. 18xx Acronyms

ADC	Analog-to-Digital Convertor
AES	Advanced Encryption Standard
CRC	Cyclic Redundancy Check
MCAN	Controller Area Network
DCC	Digital Clock Comparator
DFE	Digital Front End
DMA	Direct Memory Access
DMM	Data Modification Module
ECC	Error Correcting Code
EDMA	Enhanced Direct Memory Access
EMC	Extended Memory Controller
ePWM	Enhanced Pulse Width Modulator
ESM	Error Signaling Module

Table 2-18. 18xx Acronyms (continued)

GIO	General Input/Output
GPADC	General Purpose Analog-to-Digital Convertor
HS-RAM	Handshake RAM
I2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
L1D	Level 1 Data Memory Controller
L1P	Level 1 Program Memory Controller
L2	Level 2
L3	Level 3
LBIST	Logic Built-In Self-Test
LVDS	Low Voltage Differential Signaling
MIBSPI	Multi-Buffered Serial Peripheral Interface Module
OSC	Oscillator
PBIST	Programmable Built-In Self-Test
PKA	Public Key Algorithm
PLL	Phase Locked Loops
PRCM	Power, Reset, Clock Management
QSPI	Quad Serial Peripheral Interface
RTI	Real Time Interrupt
S/W Buffer	Software Buffer
SCI	Serial Communication Interface
SHA	Secure Hash Algorithm
STC	Self-Test Controller
TRNG	True Random Number Generator
UMC	Unified Memory Controller
VIM	Vectored Interrupt Manager

2.4.2.2 Radar Subsystem

The RADAR subsystem is responsible for the RF and analog functionality of the device. The subsystem incorporates a built-in self-test processor for the continuous motoring and calibration of the analog and RF modules. The subsystem consists of:

- FMCW transceiver
  - Integrated PLL, transmitter, receiver, baseband, and A2D
  - 76-81-GHz coverage with 4-GHz available bandwidth
  - Four receive channels
  - Three transmit channels
  - Ultra-accurate chirp engine based on fractional-n PLL
  - 12,14, or 16-bit complex analog to digital converter
- Radio processor for built-in calibration and self-test
  - ARM Cortex R4F-based radio control system
  - Built-in firmware (ROM)
  - Self-calibrating system across frequency and temperature

This subsystem is TI-programmed with an API interface to the on-chip Cortex-R4F application processor.

2.4.2.3 DSP Subsystem

The DSP subsystem consists of the following:

- The TMS320C674x™ VLIW DSP core from the generation, and the TMS320C64x+™ DSP architecture for RADAR data processing. These are enhancements from TI's C64x+™ DSP architecture, with additional features.
- 32KB L1D and 32KB of L1P cache/RAM
- 256KB of L2 RAM
- On-chip L3 shared memory of 1024KB, with 512KB dedicated to DSP and 512 KB of memory shared between the DSP and master subsystems.
- 32KB of memory for storing ADC samples from the RADAR subsystem
- Multiple Enhanced Direct Memory Access (EDMA) engines – TPCCs for high-performance data transfers
- 2-lane LVDS interface with support of up to 900 Mbps per lane for the RADAR raw ADC data transfer
- One watchdog timer and a general purpose timer implemented by the real time interrupt (RTI) modules
- One serial communication Interface (SCI) module implementing standard universal asynchronous receiver-transmitter (UART).
- Emulation capabilities
- Little Endian

2.4.2.4 Master Subsystem

The master subsystem consists of the following features:

- Cortex-R4F core supporting ARMv7-R, VFPv3-D16, and ARMv7 debug architecture
- Tightly-coupled memories
  - 128 KB of ROM
  - 512KB of program RAM with ECC
  - 192KB of data RAM with ECC
- Hardware auto-initialization of the memories
- Vectored interrupt manager for prioritizing and controlling the interrupts for different sources

2.4.2.4.1 Serial Interfaces

- One MCAN controller supporting bit rates of up to 10 Mbit/s. compliant to the controller area network (CAN) 2.0 part A, B protocol specification and ISO 11898-1, and CAN FD® V1.0 specification with up to 64 data bytes support.
- One I2C controller module with rates up to 400 kbps
- Two MIBSPI modules
- Two serial communication interface (SCI) modules implementing standard universal asynchronous receiver-transmitter (UART) with baud rates of up to 3.125 Mbps
- One quad SPI module support with maximum rate of 40 MHz

2.4.2.4.2 System Peripherals

- Multiple general-purpose input/output (GPIO) modules
- Direct memory access modules for high-performance data transfers
- One watchdog timer and a general purpose timer implemented by the real-time interrupt (RTI) modules
- Mailbox module for interprocessor communication
- Two data modification modules (DMM) with up to 65 Mbit/s data rate per pin
- Three enhanced pulse width (ePWM) modulator modules
- System reset and control module, which contains registers for the following functions:
  - Status
  - Efuse logic
  - I/O configuration

- PAD configuration
- System boot decoding logic

2.4.2.5 Functional Safety Deliverables

See the Device Safety Manual for supported features.

2.4.2.6 On-Chip Debug Support

The on-chip debug support has the following features:

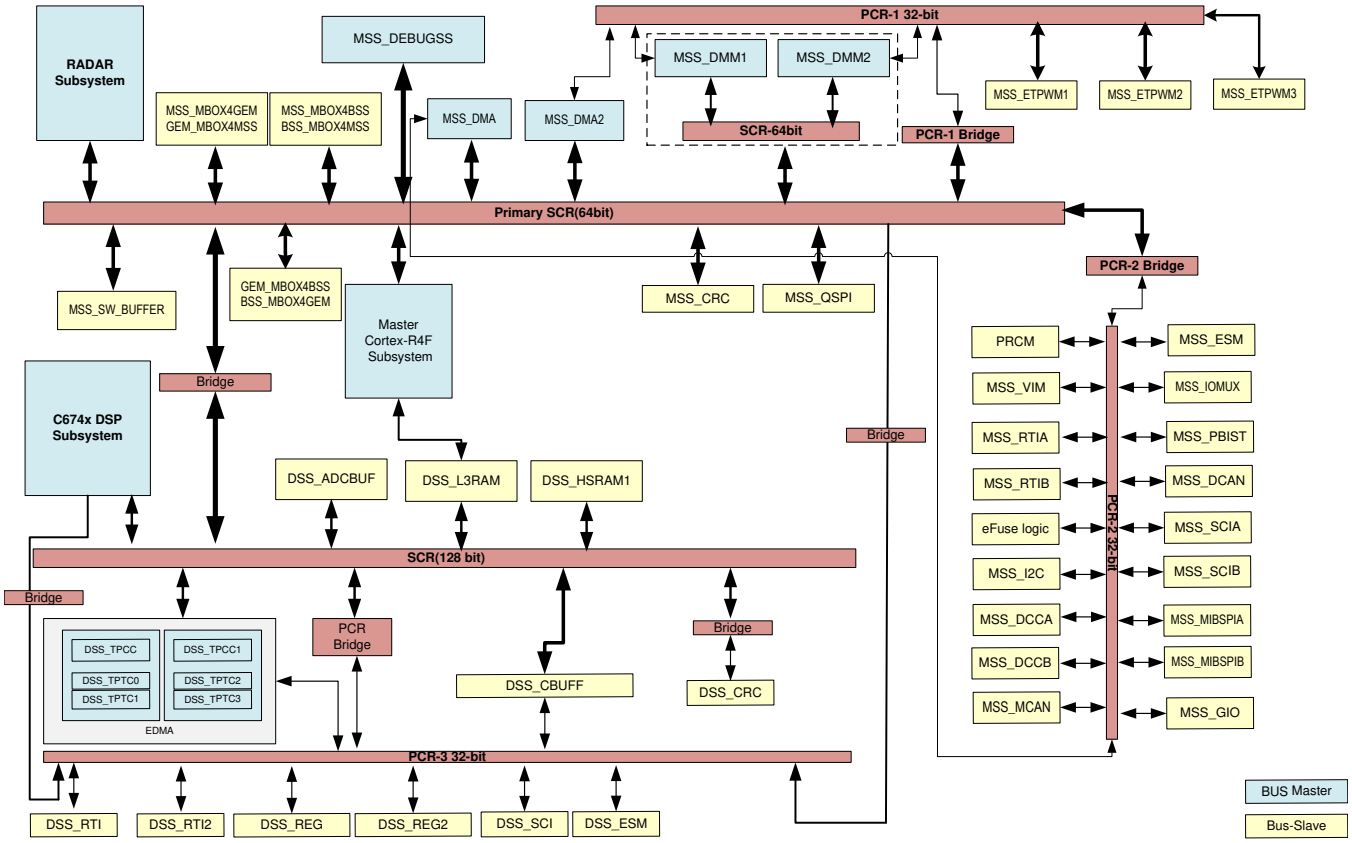
- Multiprocessor debugging to let users control multiple cores embedded in the device, such as:
  - Global starting and stopping of individual or multiple processors
  - Each processor can generate triggers to alter the execution flow of other processors
  - Interconnection of multiple devices
  - Channel triggering
- The following device cores can be debugged through Code Composer Studio (CCS):
  - Cortex-R4F
  - DSP
- Target debugging using IEEE1149.1 (JTAG®) port
- The debug subsystem includes:
  - IEEE1149.7 adapter
  - Generic TAP for emulation and test control (ICEPick-D™)
  - Debug access port (DAP)
  - Embedded trace macro (ETM)
  - Trace port interface Unit (TPIU)
  - Embedded trace buffer (ETB)

2.5 18xx Memory Map

2.5.1 System Interconnect

The device implements a system interconnect based on TI's common bus architecture, comprising of VBUSM and VBUSP protocols. Figure 2-18 shows the interconnect diagram.

Figure 2-18. System Interconnect



The system interconnect is designed for the high-performance needs of the system. Its divided into three interconnect systems local to each of the three subsystems: the RADAR subsystem, DSP subsystem, and master subsystem. The interconnection of all these subsystems is shown in Figure 3-2.

In the master subsystem, the primary VBUSM SCR is responsible for managing the arbitration priority between accesses from multiple masters to each of the slaves. The arbitration priority is always round-robin.

The master subsystem has PCR interconnect that manages the accesses to the peripheral registers and peripheral memories, and provides a global reset for all peripherals. It also supports the capability to selectively enable or disable the clock for each peripheral individually. The PCR also manages the accesses to the system module registers required to configure the device clocks, interrupts, and so forth. The system module registers include status flags for indicating exception conditions – resets, aborts, errors, and interrupts.

Similarly, the 128-bit VBUSM SCR in the DSP subsystem manages the arbitration between accesses from the multiple masters to the slaves. The DSP subsystem has a 32-bit VBUSP PCR for the system and non-system peripherals.

2.5.2 Master Subsystem Cortex-R4F Memory Map

Table 3-2 shows the master subsystem, Cortex-R4F memory map.

Table 2-19. Master Subsystem, Cortex-R4F Memory Map

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_TCMA_ROM	0x0000_0000	0x0001_7FFF	128KiB	MSS_TCMA_ROM (TCMA) Program ROM (refer to ROM Eclipsing section)

Table 2-19. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
Reserved	0x0001_8000	0x001F_FFFF		Reserved (refer to ROM Eclipsing section)
MSS_TCMA_RAM	0x0020_0000	0x07FF_FFFF	512KiB	MSS_TCMA_RAM (TCMA) size varies based on device and DSS_L3 (L3) sharing options configured (refer to ROM Eclipsing section)
MSS_TCMB	0x0800_0000	0x0C1F_FFFF	192KiB	MSS_TCMB (TCMB) Data RAM
MSS_SW_BUFFER	0x0C20_0000	0x0C20_1FFF	8KiB	MSS_SW_BUFFER (SWBUFF) Scratchpad memory
Reserved	0x0C20_2000	0x4FFF_FFFF		Reserved
DSS_TPTC0	0x5000_0000	0x5000_03FF	792B	DSS_TPTC0 (EDMA TPTC0) module configuration space
DSS_REG	0x5000_0400	0x5000_07FF	864B	DSS_REG (DSPSS) control module registers
DSS_TPTC1	0x5000_0800	0x5000_0BFF	792B	DSS_TPTC1 (EDMA TPTC1) module configuration space
DSS_REG2	0x5000_0C00	0x5000_FFFF	676B	DSS_REG2 (DSPSS) control module registers
DSS_TPCC	0x5001_0000	0x5001_FFFF	16KiB	DSS_TPCC (EDMA TPCC0) module configuration space
DSS_RTI	0x5002_0000	0x5002_FFFF	192B	DSS_RTI (WDT/RTI1) configuration space
DSS_SCI	0x5003_0000	0x5003_FFFF	148B	DSS_SCI (SCI) memory space
DSS_STC	0x5004_0000	0x5004_FFFF	284B	DSS_STC (STC) module configuration space (refer to Safety chapter)
Reserved	0x5005_0000	0x5006_FFFF		Reserved
DSS_CBUFF	0x5007_0000	0x5007_FFFF	564B	DSS_CBUFF (CBUFF) module configuration registers (refer to HSI chapter)
DSS_HW_ACC_PARAM	0x5008_0000	0x5008_07FF	512B	DSS_HW_ACC_PARAM (HWA) FFT accelerator PARAM memory
DSS_HW_ACC_STATIC	0x5008_0800	0x5008_0FFF	616B	DSS_HW_ACC_STATIC (HWA) FFT accelerator configuration registers
DSS_HW_ACC_WIN	0x5008_1000	0x5008_FFFF	4KiB	DSS_HW_ACC_WIN (HWA) FFT accelerator Window registers
DSS_TPTC2	0x5009_0000	0x5009_03FF	792B	DSS_TPTC2 (EDMA TPTC2) module configuration space
DSS_TPTC3	0x5009_0400	0x5009_FFFF	792B	DSS_TPTC3 (EDMA TPTC3) module configuration space
DSS_TPCC1	0x500A_0000	0x500A_FFFF	16KiB	DSS_TPCC1 (EDMA TPCC1) module configuration space
Reserved	0x500B_0000	0x500C_FFFF		Reserved

Table 2-19. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_ESM	0x500D_0000	0x500E_FFFF	1KiB	DSS_ESM (ESM) module configuration registers (refer to Safety chapter)
DSS_RTI2	0x500F_0000	0x500F_FFFF	192B	DSS_RTI2 (RTI2) module configuration registers
Reserved	0x5010_0000	0x50FF_FFFF		Reserved
DSS_L3RAM	0x5100_0000	0x51FF_FFFF	2MB	DSS_L3RAM (L3) shared memory space
DSS_ADCBUF	0x5200_0000	0x5201_FFFF	32KiB	DSS_ADCBUF (ADC) buffer memory space
DSS_CBUFF_FIFO	0x5202_0000	0x5202_7FFF	16KiB	DSS_CBUFF_FIFO (CBUFF) FIFO space (refer to HSI chapter)
Reserved	0x5202_8000	0x5202_FFFF		Reserved
DSS_FFT_ACC_DMA1	0x5203_0000	0x5203_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator Memory -1 space
DSS_FFT_ACC_DMA2	0x5203_8000	0x5206_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator Memory -2 space
Reserved	0x5207_0000	0x5207_FFFF		
DSS_HSRAM1	0x5208_0000	0x5208_FFFF	32KiB	DSS_HSRAM1 (HSRAM) Handshake memory space
Reserved	0x5209_0000	0x577D_FFFF		Reserved
DSS_DSP_L2_UMAP1	0x577E_0000	0x577F_FFFF	128KiB	DSS_DSP_L2_UMAP1 (L2) RAM space
DSS_DSP_L2_UMAP0	0x5780_0000	0x57DF_FFFF	128KiB	DSS_DSP_L2_UMAP0 (L2) RAM space
DSS_DSP_L1P	0x57E0_0000	0x57EF_FFFF	32KiB	DSS_DSP_L1P (L1) program memory space
DSS_DSP_L1D	0x57F0_0000	0xBFFF_FFFF	32KiB	DSS_DSP_L1D (L1) data memory space
EXT_FLASH	0xC000_0000	0xC07F_FFFF	8MB	MSS_QSPI (QSPI) flash memory space
MSS_QSPI	0xC080_0000	0xF060_0FFF	116B	MSS_QSPI (QSPI) module configuration registers
MSS_MBOX4BSS	0xF060_1000	0xF060_1FFF	2KiB	MSS_MBOX4BSS mailbox memory space
BSS_MBOX4MSS	0xF060_2000	0xF060_3FFF	2KiB	BSS_MBOX4MSS mailbox memory space
GEM_MBOX4MSS	0xF060_4000	0xF060_4FFF	2KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0xF060_5000	0xF060_5FFF	2KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0xF060_6000	0xF060_6FFF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0xF060_7000	0xF060_7FFF	2KiB	BSS_MBOX4GEM mailbox memory space
BSS_MBOX4MSS_REG	0xF060_8000	0xF060_80FF	188B	BSS_MBOX4MSS_REG mailbox Configuration registers
BSS_MBOX4GEM_REG	0xF060_8100	0xF060_81FF	188B	BSS_MBOX4GEM_REG mailbox Configuration registers

Table 2-19. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
GEM_MBOX4BSS_REG	0xF060_8200	0xF060_82FF	188B	GEM_MBOX4BSS_REG mailbox Configuration registers
MSS_MBOX4GEM_REG	0xF060_8300	0xF060_83FF	188B	MSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4MSS_REG	0xF060_8400	0xF060_85FF	188B	GEM_MBOX4MSS_REG mailbox Configuration registers
MSS_MBOX4BSS_REG	0xF060_8600	0xFCF7_8BFF	188B	MSS_MBOX4BSS_REG mailbox Configuration registers
MSS_ETPWM1	0xFCF7_8C00	0xFCF7_8CFF	116B	MSS_ETPWM1 (ePWM1) configuration registers
MSS_ETPWM2	0xFCF7_8D00	0xFCF7_8DFF	116B	MSS_ETPWM2 (ePWM2) configuration registers
MSS_ETPWM3	0xFCF7_8E00	0xFCF8_0FFF	116B	MSS_ETPWM3 (ePWM3) configuration registers
MSS_DMA2_RAM	0xFCF8_1000	0xFCFF_0FFF	4KiB	MSS_DMA2_RAM (DMA2) RAM memory space
MSS_PCR2	0xFCFF_1000	0xFCFF_F5FF	1KiB	MSS_PCR2 (PCR_2) interconnect configuration port
MSS_DMM2	0xFCFF_F600	0xFCFF_F6FF	472B	MSS_DMM2 (DMM2) module configuration registers
MSS_DMM	0xFCFF_F700	0xFCFF_F7FF	472B	MSS_DMM (DMM1) module configuration registers
MSS_DMA2_REG	0xFCFF_F800	0xFCFF_FFFF	1KiB	MSS_DMA2_REG (DMA2) module configuration registers
MSS_DTHE	0xFD00_0000	0xFD00_3FFF	3KiB	MSS_DTHE (Crypto) module configuration registers
Reserved	0xFD00_4000	0xFDFF_FFFF		Reserved
MSS_MCRC	0xFE00_0000	0xFF0B_FFFF	16KiB	MSS_MCRC (CRC) module configuration registers
MSS_MIBSPIB_TXRAM	0xFF0C_0000	0xFF0C_01FF	0.5KiB	MSS_MIBSPIB_TXRAM (MIBSPIB) TX RAM memory space
MSS_MIBSPIB_RXRAM	0xFF0C_0200	0xFF0D_FFFF	0.5KiB	MSS_MIBSPIB_RXRAM (MIBSPIB) RX RAM memory space
MSS_MIBSPIA_TXRAM	0xFF0E_0000	0xFF0E_01FF	0.5KiB	MSS_MIBSPIA_TXRAM (MIBSPIA) TX RAM memory space
MSS_MIBSPIA_RXRAM	0xFF0E_0200	0xFF1D_FFFF	0.5KiB	MSS_MIBSPIA_RXRAM (MIBSPIA) RX RAM memory space
Reserved	0xFF1E_0000	0xFF4F_FFFF		Reserved
MSS_MCAN_MSGMEM	0xFF50_0000	0xFF9F_FFFF	68KiB	MSS_MCAN_MSGMEM (MCAN) RAM memory space
MSS_DEBUGSS	0xFFA0_0000	0xFFF7_7FFF	244KiB	MSS_DEBUGSS (Debug subsystem) memory space and registers

Table 2-19. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_PCR	0xFFF7_8000	0xFFF7_9FFF	1KiB	MSS_PCR (PCR_1) interconnect configuration port
MSS_MCAN_ECC	0xFFF7_A000	0xFFF7_BBFF	452B	MSS_MCAN_ECC (MCAN) module registers
MSS_GIO	0xFFF7_BC00	0xFFF7_C7FF	180B	MSS_GIO (GIO) module configuration registers
MSS_MCAN_CFG	0xFFF7_C800	0xFFF7_D3FF	768B	MSS_MCAN_CFG (MCAN) module configuration registers
MSS_I2C	0xFFF7_D400	0xFFF7_DBFF	112B	MSS_I2C (I2C) module configuration registers
Reserved	0xFFF7_DC00	0xFFF7_E4FF		Reserved
MSS_SCIA	0xFFF7_E500	0xFFF7_E6FF	148B	MSS_SCIA (SCIA/UART) module configuration registers
MSS_SCIB	0xFFF7_E700	0xFFF7_F3FF	148B	MSS_SCIB (SCIB/UART) module configuration registers
MSS_MIBSPIA	0xFFF7_F400	0xFFF7_F5FF	512B	MSS_MIBSPIA (MIBSPIA) module configuration registers
MSS_MIBSPIB	0xFFF7_F600	0xFFF7_FFFF	512B	MSS_MIBSPIB (MIBSPIB) module configuration registers
MSS_DMA_RAM	0xFFF8_0000	0xFFF8_1FFF	4KiB	MSS_DMA_RAM (DMA1) RAM memory space
MSS_VIM_MEM	0xFFF8_2000	0xFFF8_BFFF	2KB	MSS_VIM_MEM (VIM) RAM memory space
Reserved	0xFFF8_C000	0xFFFF_E0FF		Reserved
MSS_TOPRCM	0xFFFF_E100	0xFFFF_E3FF	756B	MSS_TOPRCM TOP Level Reset, Clock management registers
MSS_PBIST	0xFFFF_E400	0xFFFF_E5FF	464B	MSS_PBIST (PBIST) module configuration registers (refer to Safety chapter)
MSS_STC	0xFFFF_E600	0xFFFF_E9FF	284B	MSS_STC (STC) module configuration registers (refer to Safety chapter)
MSS_IOMUX	0xFFFF_EA00	0xFFFF_EBFF	512B	MSS_IOMUX (IOMUX) module registers
MSS_DCCA	0xFFFF_EC00	0xFFFF_EDFF	44B	MSS_DCCA (DCCA) module configuration registers
MSS_RTIB	0xFFFF_EE00	0xFFFF_EFFF	192B	MSS_RTIB (WDT/RTIB) module configuration registers
MSS_DMA_REG	0xFFFF_F000	0xFFFF_F3FF	1KiB	MSS_DMA_REG (DMA1) module configuration registers
MSS_DCCB	0xFFFF_F400	0xFFFF_F4FF	44B	MSS_DCCB (DCCB) module configuration registers
MSS_ESM	0xFFFF_F500	0xFFFF_F5FF	156B	MSS_ESM (ESM) module configuration registers (refer to Safety chapter)

Table 2-19. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
Reserved	0xFFFF_F600	0xFFFF_F7FF		Reserved
MSS_GPCFG_REG	0xFFFF_F800	0xFFFF_FBFF	352B	MSS_GPCFG_REG (GPCFG) General purpose control registers
MSS_RTIA	0xFFFF_FC00	0xFFFF_FCFF	192B	MSS_RTIA (RTIA) module configuration registers
MSS_VIM	0xFFFF_FD00	0xFFFF_FEFF	512B	MSS_VIM (VIM) module configuration registers
MSS_RCM	0xFFFF_FF00	0xFFFF_FFFF	256B	MSS_RCM (RCM) Reset, Clock management registers

2.5.2.1 Radar Subsystem Interface

The RADAR subsystem is accessible through a set of TI-implemented high-level API calls by the application running on the master CR4F. For more information on the 18xx RADAR subsystem interface, see the 18xx Interface control document.

2.5.3 DSP Subsystem Memory Map

Table 3-3 shows the DSP C674x memory map.

Table 2-20. DSP C674x Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSP_L2_UMAP1	0x007E_0000	0x007F_FFFF	128KiB	DSP_L2_UMAP1 (L2) RAM space
DSP_L2_UMAP0	0x0080_0000	0x0081_FFFF	128KiB	DSP_L2_UMAP0 (L2) RAM space
DSP_L1P	0x00E0_0000	0x00E0_7FFF	32KiB	DSP_L1P (L1) program memory space
DSP_L1D	0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D (L1) data memory space
DSS_TPTC0	0x0200_0000	0x0200_03FF	1KiB	DSS_TPTC0 (EDMA TPTC0) module configuration space
DSS_REG	0x0200_0400	0x0200_07FF	864B	DSS_REG (DSPSS) control module registers
DSS_TPTC1	0x0200_0800	0x0200_0BFF	1KiB	DSS_TPTC1 (EDMA TPTC1) module configuration space
DSS_REG2	0x0200_0C00	0x0200_0FFF	624B	DSS_REG2 (DSPSS) control module registers
DSS_TPCC	0x0201_0000	0x0201_3FFF	16KiB	DSS_TPCC (EDMA TPCC0) module configuration space
DSS_RTI	0x0202_0000	0x0202_00FF	192B	DSS_RTI (WDT/RTI1) module configuration registers
DSS_SCI	0x0203_0000	0x0203_00FF	148B	DSS_SCI (SCI/UART) module Configuration registers
DSS_CBUFF	0x0207_0000	0x0207_03FF	564B	DSS_CBUFF (CBUFF) module Configuration registers (refer to HSI chapter)
DSS_HW_ACC_PARAM	0x0208_0000	0x0208_07FF	512B	DSS_HW_ACC_PARAM (HWA) FFT accelerator PARAM memory
DSS_HW_ACC_STATIC	0x0208_0800	0x0208_0FFF	616B	DSS_HW_ACC_STATIC (HWA) FFT accelerator configuration registers
DSS_HW_ACC_WIN	0x0208_1000	0x0208_FFFF	4KiB	DSS_HW_ACC_WIN (HWA) FFT accelerator Window registers
DSS_TPTC2	0x0209_0000	0x0209_03FF	1KiB	DSS_TPTC2 (EDMA TPTC2) module configuration space
DSS_TPTC3	0x0209_0400	0x0209_07FF	1KiB	DSS_TPTC3 (EDMA TPTC3) module configuration space

Table 2-20. DSP C674x Memory Map (continued)

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_TPCC1	0x020A_0000	0x020A_3FFF	16KiB	DSS_TPCC1 (EDMA TPCC1) module configuration space
Reserved	0x020B_0000	0x020C_FFFF		Reserved
DSS_ESM	0x020D_0000	0x020E_FFFF	92B	DSS_ESM (ESM) module Configuration registers (refer to Safety chapter)
DSS_RTI2	0x020F_0000	0x020F_00FF	192B	DSS_RTI2 (RTI2) module configuration registers
DSS_FFT_ACC_DMA1	0x2103_0000	0x2103_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator Memory -1 space
DSS_FFT_ACC_DMA2	0x2103_8000	0x2103_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator Memory -2 space
Reserved	0x0210_0000	0x0460_7FFF		Reserved
BSS_MBOX4MSS_REG	0x0460_8000	0x0460_80FF	188B	BSS_MBOX4MSS_REG mailbox Configuration registers
BSS_MBOX4GEM_REG	0x0460_8100	0x0460_81FF	188B	BSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4BSS_REG	0x0460_8200	0x0460_82FF	188B	GEM_MBOX4BSS_REG mailbox Configuration registers
MSS_MBOX4GEM_REG	0x0460_8300	0x0460_83FF	188B	MSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4MSS_REG	0x0460_8400	0x0460_84FF	188B	GEM_MBOX4MSS_REG mailbox Configuration registers
MSS_MBOX4BSS_REG	0x0460_8600	0x0460_86FF	188B	MSS_MBOX4BSS_REG mailbox Configuration registers
Reserved	0x050C_0000	0x1FFF_FFFF		Reserved
DSS_L3RAM	0x2000_0000	0x201F_FFFF	2MB	DSS_L3RAM (L3) shared memory space
DSS_ADCBUF	0x2100_0000	0x2100_7FFC	32KiB	DSS_ADCBUF (ADC buffer) memory space
DSS_CBUFF_FIFO	0x2102_0000	0x2102_3FFC	16KiB	DSS_CBUFF_FIFO (Common buffer) FIFO space (Refer to HSI chapter)
Reserved	0x2102_8000	0x2107_FFFF		Reserved
DSS_HSRAM1	0x2108_0000	0x2108_7FFC	32KiB	DSS_HSRAM1 (HSRAM) Handshake memory space
Reserved	0x2109_0000	0x21FF_FFFF		Reserved
DSS_MCRC	0x2200_0000	0x2200_03FF	1KiB	DSS_MCRC (CRC) module Configuration registers
Reserved	0x2500_0000	0x5060_0FFF		Reserved
MSS_MBOX4BSS	0x5060_1000	0x5060_17FF	2KiB	MSS_MBOX4BSS mailbox memory space
BSS_MBOX4MSS	0x5060_2000	0x5060_27FF	2KiB	BSS_MBOX4MSS mailbox memory space
GEM_MBOX4MSS	0x5060_4000	0x5060_47FF	2KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0x5060_5000	0x5060_57FF	2KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0x5060_6000	0x5060_67FF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_57FF	2KiB	BSS_MBOX4GEM mailbox memory space
Reserved	0x5600_0000	0x5060_5FFF		Reserved
GEM_MBOX4BSS	0x5060_6000	0x5060_67FF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_7FFF	2KiB	BSS_MBOX4GEM mailbox memory space
Reserved	0x5600_0000	0xFFFF_FFFF		Reserved

2.5.4 EDMA Memory Map

Table 3-4 shows the EDMA-TPTC memory map.

Table 2-21. EDMA-TPTC Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_EDMA_SCI	0x0603_0000	0x0603_00FF	148B	DSS_SCI memory space view from EDMA
DSS_DSP_L2_UMAP1	0x107E_0000	0x107F_FFFF	128KiB	DSP_L2_UMAP1 (L2) memory view from EDMA
DSS_DSP_L2_UMAP0	0x1080_0000	0x1081_FFFF	128KiB	DSP_L2_UMAP0 (L2) memory view from EDMA
DSS_DSP_L1P	0x10E0_0000	0x10E0_7FFF	32KiB	DSP_L1P (L1) program memory view from EDMA
DSS_DSP_L1D	0x10F0_0000	0x10F0_7FFF	32KiB	DSP_L1D (L1) data memory view from EDMA
DSS_L3RAM	0x2000_0000	0x201F_FFFF	2MB	DSS_L3RAM shared memory space
DSS_ADCBUF	0x2100_0000	0x2100_7FFC	32KiB	DSS_ADCBUF memory space
DSS_CBUFF_FIFO	0x2102_0000	0x2102_3FFC	16KiB	DSS_CBUFF_FIFO (Common buffer) memory space (Refer to HSI chapter)
DSS_FFT_ACC_DMA1	0x2103 0000	0x2103_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator memory-1 space
DSS_FFT_ACC_DMA2	0x2103 8000	0x2107_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator memory-2 space
DSS_HSRAM1	0x2108_0000	0x2108_7FFC	32KiB	DSS_HSRAM1 (Handshake) memory
MSS_TCMA_RAM	0x4020_0000	0x4023_FFFF	256 KiB	MSS_TCMA_RAM (TCMA) Data RAM
MSS_TCMB	0x4800_0000	0x4802_FFFF	192 KiB	MSS_TCMB (TCMB) Data RAM
MSS_SW_BUFFER	0x4C20_0000	0x4C20_1FFF	8 KiB	MSS_SW_BUFFER S/W Scratchpad memory
GEM_MBOX4MSS	0x5060_4000	0x5060_4000	2 KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0x5060_5000	0x5060_5000	2 KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0x5060_6000	0x5060_6000	2 KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_7000	2 KiB	BSS_MBOX4GEM mailbox memory space

2.6 18xx Integration

2.6.1 Cortex-R4F Subsystem

2.6.1.1 Tightly Coupled Memories

Table 3-5 lists the dedicated MSS_TCMA_RAM and MSS_TCMB sizes for the Cortex R4F processor in the master subsystem, and also mentions the total available L3 shared RAM in the device. A portion of this L3 shared memory (DSS_L3RAM) can be allotted as TCM, to further increase the MSS_TCMA_RAM and MSS_TCMB available for the Cortex R4F.

Table 2-22. TCM and Shared Memory Available for Cortex R4F in Master Subsystem

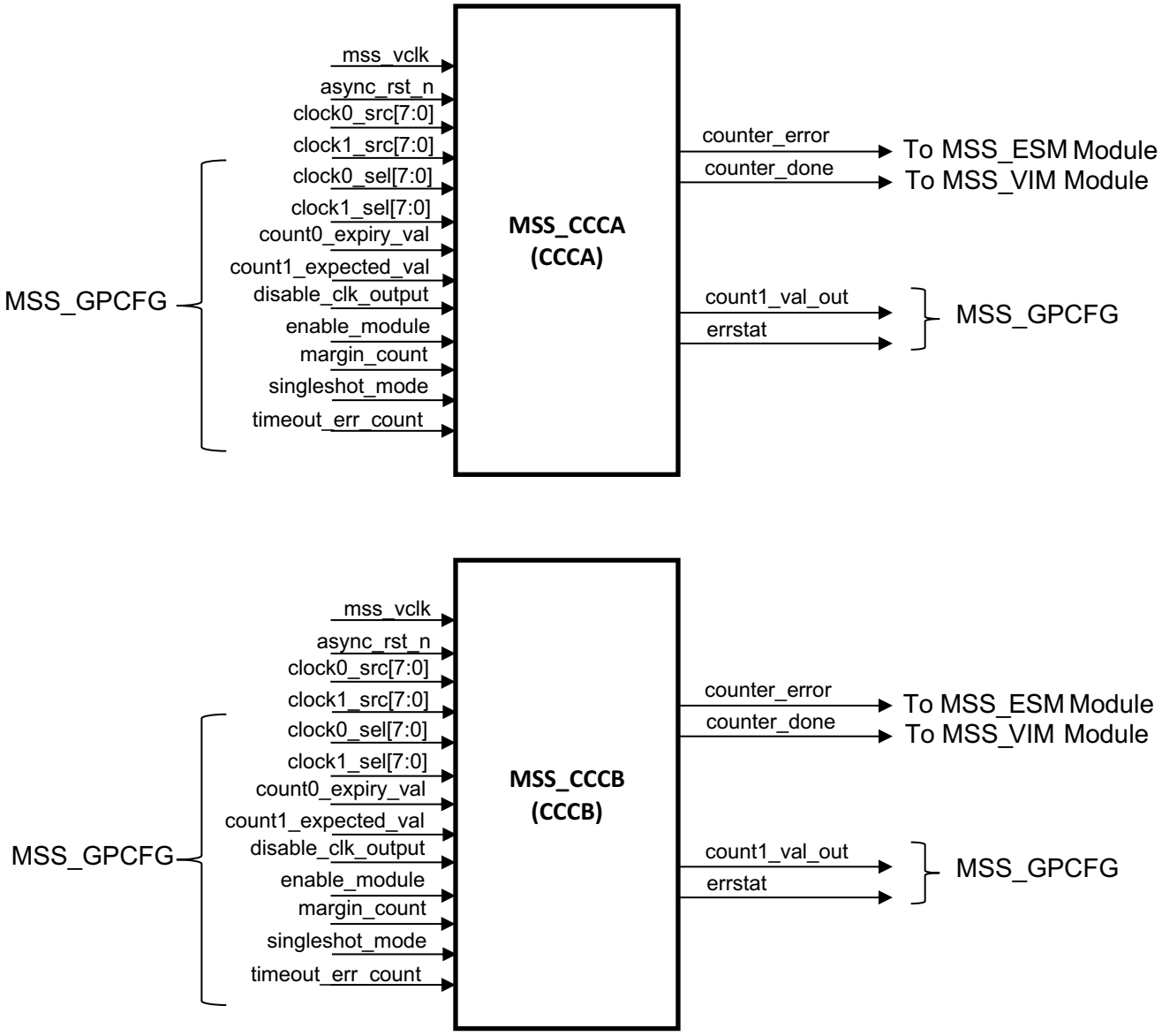
Cortex R4F		Shared
MSS_TCMA_RAM (Program RAM) (KB)	MSS_TCMB (Data RAM) (KB)	L3 Shared (KB)
512	192	1024

See on how the L3 shared memory (DSS_L3RAM) can be assigned between the Cortex R4F of the master subsystem and the DSP core.

2.6.2 Clock Comparator

2.6.2.1 Core Clock Comparator (MSS_CCCA/MSS_CCCB)

Figure 2-19. Integration of MSS_CCCA and MSS_CCCB Modules



2.6.2.1.1 MSS_CCCA and MSS_CCCB Integration Connections

This device has two instances of CCC: MSS_CCCA (CCCA) and MSS_CCCB (CCCB). The clock connectivity information for these two instances is provided in Table 3-6. Configuration and status of this module is available through the MSS_GPCFG registers of the device.

Table 2-23. MSS_CCCA and MSS_CCCB Integration Connections

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
counter_error	ESM_GRP1[1]	ESM_GRP1[4]
counter_done	IRQ[80]	IRQ[81]

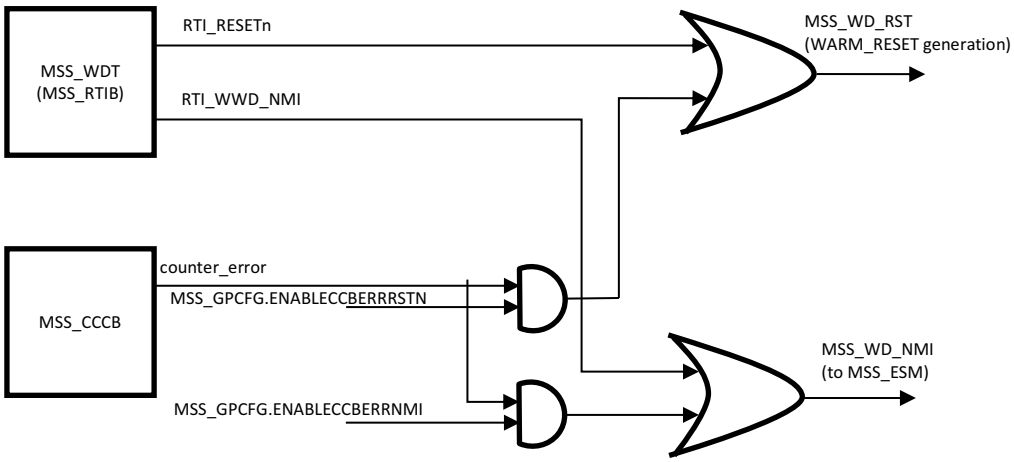
Table 2-23. MSS_CCCA and MSS_CCCB Integration Connections (continued)

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
clock0_src[0]	REFCLK	CR4_VCLK
clock0_src[1]	CPUCLK	DSSCLK
clock0_src[2]	RCCLK	BSSCLK
clock0_src[3]	RCCLK	QSPICLK
clock0_src[4]	RCCLK	N/A
clock0_src[5]	RCCLK	REFCLK
clock0_src[6]	RCCLK	CPUCLK
clock0_src[7]	RCCLK	RCCLK
clock1_src[0]	REFCLK	PLLCLK_600
clock1_src[1]	PLLCLK_600	MSS_VCLK
clock1_src[2]	PLL 240Mhz	CPUCLK
clock1_src[3]	RCCLK	CR4_VCLK
clock1_src[4]	RCCLK	MSS_VCLK
clock1_src[5]	RCCLK	DSSCLK
clock1_src[6]	RCCLK	BSSCLK
clock1_src[7]	RCCLK	QSPICLK

2.6.2.1.2 MSS_CCCB Integration to MSS_WD

As a safety requirement, WDT IP should work on an independent clock source instead of the clock used by the MSS CR4. Because the WDT IP does not allow this flexibility, an additional monitoring logic is added in the form of CCM (MSS_CCCB instance), coupled along with the watchdog. MSS_CCCB is used to compare CR4_VCLK to an independent reference clock, such as XTAL. If the CR4 clock indicates a deviation from the expected frequency, a WD reset or a WD NMI can be issued.

Figure 2-20. MSS_CCCB Integration to MSS_WD



2.6.2.2 Dual Clock Comparator (MSS_DCCA/MSS_DCCB)

Figure 2-21. MSS_DCCA/MSS_DCCB Integration Diagram

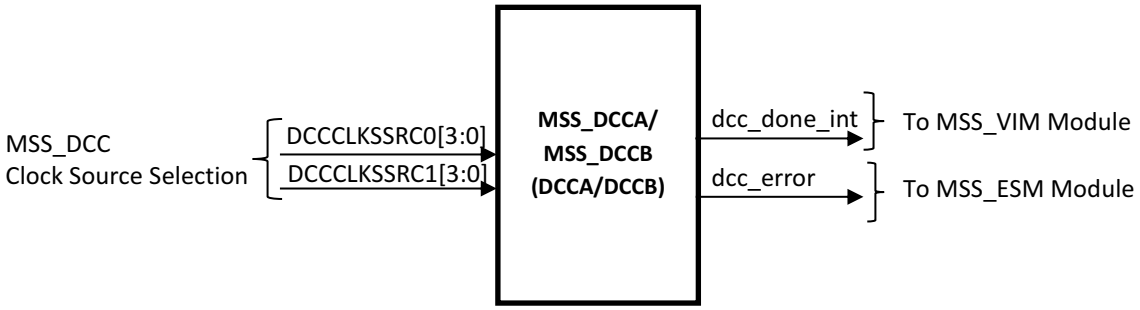


Table 2-24. MSS_DCCA Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - REF_CLK	0x0 - REF_CLK
0xA - PLL_600	0x1 - CPU_CLK
0x5 - PLL_240	0x2 to 0x7 - RC_CLK

Table 2-25. MSS_DCCB Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - PLL_600	0x0 - VCLK
0xA - VCLK	0x1 - DSS_CLK
0x5 - CPU_CLK	0x2 - BSS_CLK
	0x3 - QSPI_CLK
	0x4 - Reserved
	0x5 - CPU_CLK
	0x6 - REF_CLK
	0x7 - RC_CLK

NOTE: Any values not mentioned are not used.

2.6.3 C674x DSP Subsystem

2.6.3.1 DSP Event Assignment

Table 2-26. DSP Event Assignment

Event No.	Interrupt	Description
0	EVT0	Output of event combiner0, for events 1 through 31
1	EVT1	Output of event combiner0, for events 32 through 63
2	EVT2	Output of event combiner0, for events 64 through 95
3	EVT3	Output of event combiner0, for events 96 through 127
4	Reserved	Reserved
5	Reserved	Reserved
6	Reserved	Reserved
7	Reserved	Reserved
8	Reserved	Reserved
9	Reserved	Reserved

Table 2-26. DSP Event Assignment (continued)

Event No.	Interrupt	Description
10	Reserved	Reserved
11	Reserved	Reserved
12	Reserved	Reserved
13	IDMAINT0	From DSP EMC, IDMA Channel 0 Interrupt
14	IDMAINT1	From DSP EMC, IDMA Channel 1Interrupt
15	Reserved	Reserved
16	DSS_TPTC0_IRQ_DONE	DSS_TPTC0 (EDMA TPTC0) completion interrupt
17	DSS_TPTC0_IRQ_ERR	DSS_TPTC0 (EDMA TPTC0) Error Interrupt
18	DSS_TPTC1_IRQ_DONE	DSS_TPTC1 (EDMA TPTC1) completion interrupt
19	DSS_TPTC1_IRQ_ERR	DSS_TPTC1 (EDMA TPTC1) Error Interrupt
20	DSS_TPCC_IRQ_DONE	DSS_TPCC (EDMA TPCC0) Global completion Interrupt
21	DSS_TPCC_IRQ_ERR	DSS_TPCC (EDMA TPCC0) Error Interrupt
22	DSS_CBUFF_IRQ	DSS_CBUFF (COMMON BUFFER) Interrupt
23	Reserved	Reserved
24	DSS_CBUFF_ERR_INTR	DSS_CBUFF (COMMON BUFFER) Error Interrupt
25	Reserved	Reserved
26	DSS_FRAME_START_IRQ/DSS_DMMSWIN T0/DSS_DMMSWINT39	Mux of VIN Frame start or BSS DFE Frame start.
27	DSS_CHIRP_AVAIL_IRQ/DSS_DMMSWINT2 /DSS_DMMSWINT43	Mux of VIN Chirp Available or DFE chirp available..
28	Reserved	Reserved
29	FFT_ACC_PARAM_DONE_INTR	DSS_HW_ACC FFT accelerator - param done interrupt
30	FFT_ACC_DONE_INTR	DSS_HW_ACC FFT accelerator - done interrupt
31	FFT_ACC_ACCESS_ERR	DSS_HW_ACC FFT accelerator - access error interrupt
32	DSS_ESM_LOW_PRIORITY	MSS_ESM_IRQ (Aggregate of MSS_ESM_GP1 errors)
33	DSS_MCRC_INT	MSS_MCRC (CRC) Interrupt
34	DSS_PROG_FILT_ERR	Error interrupt from Programmable filter indicating wrong programming of filter length exceeding the allowed range.
35	GEM_WAKEUP_SOURCE_FROM_DFT	Wakeup source from DFT module.
36	DSS_STC_DONE	Done indication from DSS_STC
37	DSP_PBIST_DONE	DSP_PBIST done indication from GEM
38	Reserved	Reserved
39	Reserved	Reserved
40	Reserved	Reserved
41	Reserved	Reserved
42	Reserved	Reserved
43	Reserved	Reserved
44	Reserved	Reserved
45	Reserved	Reserved
46	DSS_DMMSWINT8	Interrupt from DSS_DMM configurable
47	DSS_DMMSWINT4	Interrupt from DSS_DMM configurable
48	Reserved	Reserved
49	Reserved	Reserved
50	Reserved	Reserved
51	Reserved	Reserved
52	Reserved	Reserved
53	Reserved	Reserved
54	Reserved	Reserved

Table 2-26. DSP Event Assignment (continued)

Event No.	Interrupt	Description
55	Reserved	Reserved
56	Reserved	Reserved
57	Reserved	Reserved
58	DSS_MSS_SW0	DSS_MSS_SW interrupt
59	DSS_MSS_SW1	DSS_MSS_SW interrupt
60	DSS_DMMSWINT5	Interrupt from DSS_DMM configurable
61	DSS_DMMSWINT6	Interrupt from DSS_DMM configurable
62	DSS_BSS_SW1	Radar SS SW Interrupt 0
63	DSS_BSS_SW2	Radar SS SW Interrupt 1
64	DSS_TPTC2_IRQ_DONE	DSS_TPTC2 (EDMA TPTC2) completion interrupt
65	DSS_TPTC2_IRQ_ERR	DSS_TPTC2 (EDMA TPTC2) Error Interrupt
66	DSS_TPTC3_IRQ_DONE	DSS_TPTC3 (EDMA TPTC3) completion interrupt
67	DSS_TPTC3_IRQ_ERR	DSS_TPTC3 (EDMA TPTC3) Error Interrupt
68	DSS_TPCC1_IRQ_DONE	DSS_TPCC1 (EDMA TPCC1) Global completion Interrupt
69	DSS_TPCC1_IRQ_ERR	DSS_TPCC1 (EDMA TPCC1) Error Interrupt
70	DSS_ADC_DATA_VALID_FALL/DSS_DMMS WINT3/DSS_DMMSWINT44	DSS_ADC Ping/Pong interrupt
71	DSS_UART_REQ0	DSS_SCI (UART) Req 0
72	DSS_UART_REQ1	DSS_SCI (UART) Req 1
73	DSS_RTI0_OVERFLOW_0	DSS_RTI Overflow 0
74	DSS_RTI0_OVERFLOW_1	DSS_RTI Overflow 1
75	DSS_RTI0_0	DSS_RTI Interrupt 0
76	DSS_RTI0_1	DSS_RTI Interrupt 1
77	DSS_RTI0_2	DSS_RTI Interrupt 2
78	DSS_RTI0_3	DSS_RTI Interrupt 3
79	DSS_RTI1_OVERFLOW_0	DSS_RTI2 Overflow 0
80	DSS_RTI1_OVERFLOW_1	DSS_RTI2 Overflow 1
81	DSS_RTI1_0	DSS_RTI2 Interrupt 0
82	DSS_RTI1_1	DSS_RTI2 Interrupt 1
83	DSS_RTI1_2	DSS_RTI2 Interrupt 2
84	DSS_RTI1_3	DSS_RTI2 Interrupt 3
85	DSS_BSS_MAILBOX_FULL	Interrupt indicating there is a message from MSS in the Mailbox BSS-DSS
86	DSS_BSS_MAILBOX_EMPTY	Interrupt indicating the MSS has read/ack the message DSP posted in the Mailbox DSS-BSS
87	GPIO_0_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
88	GPIO_1_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
89	GPIO_2_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
90	Reserved	Reserved
91	DSS_MSS_MAILBOX_FULL	Interrupt indicating there is a message from MSS in the Mailbox MSS-DSS
92	DSS_MSS_MAILBOX_EMPTY	Interrupt indicating the MSS has read/ack the message DSP posted in the Mailbox DSS-MSS
93	DSS_LOGICAL_FRAME_START/DSS_DMM SWINT1/DSS_DMMSWINT40	Logical Frame start interrupt
94	DSS_DMMSWINT7	Interrupt from DSS_DMM configurable
95	Reserved	Reserved
96	INTERR	DSP dropped CPU interrupt event
97	IDMA_ERR	Invalid IDMA parameters

Table 2-26. DSP Event Assignment (continued)

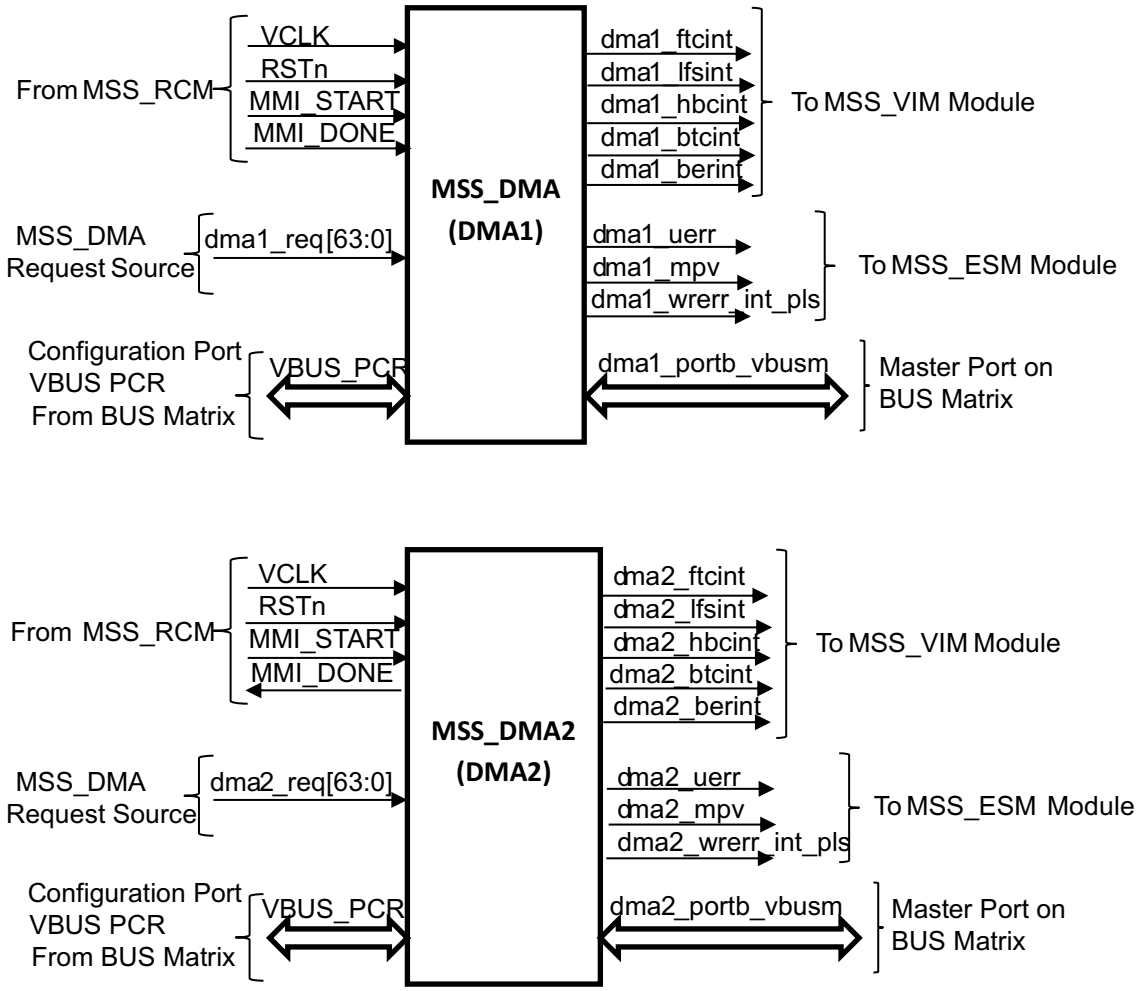
Event No.	Interrupt	Description
98	Reserved	Reserved
99	Reserved	Reserved
100	Reserved	Reserved
101	Reserved	Reserved
102	Reserved	Reserved
103	Reserved	Reserved
104	Reserved	Reserved
105	Reserved	Reserved
106	Reserved	Reserved
107	Reserved	Reserved
108	Reserved	Reserved
109	Reserved	Reserved
110	Reserved	Reserved
111	Reserved	Reserved
112	Reserved	Reserved
113	DSP_PMC_ED	DSS_DSP_L1P parity error
114	Reserved	Reserved
115	Reserved	Reserved
116	DSP_UMC_ED1	DSS_DSP_L2 ECC single error correction
117	DSP_UMC_ED2	DSS_DSP_L2 ECC double error detection
118	DSP_PDC_INT	Power down sleep interrupt
119	DSP_SYS_CMPA	CPU memory protection fault
120	DSP_L1P_CMPA	DSS_DSP_L1P CPU memory protection fault
121	DSP_L1P_DMPA	DSS_DSP_L1P DMA memory protection fault
122	DSP_L1D_CMPA	DSS_DSP_L1D CPU memory protection fault
123	DSP_L1D_DMPA	DSS_DSP_L1D DMA memory protection fault
124	DSP_L2_CMPA	DSS_DSP_L2 CPU memory protection fault
125	DSP_L2_DMPA	DSS_DSP_L2 DMA memory protection fault
126	DSP_EMC_CMPA	From EMC, CPU memory protection fault
127	DSP_EMC_BUSSERR	From EMC, bus error interrupt

2.6.4 Direct Memory Access Controller (MSS_DMA)

2.6.4.1 MSS_DMA Integration Diagrams

The device has two instances of DMA module, MSS_DMA and MSS_DMA2. Integration of the two DMA blocks in the device are shown in Figure 3-5 and .

Figure 2-22. Integration of MSS_DMA and MSS_DMA2 Module



2.6.4.2 MSS_DMA Features

- 64-bit OCP protocol to perform bus master accesses
- INCR-4 64-bit burst accesses
- Multithreading architecture allowing data of two different channel transfers to be interleaved during non-burst accesses
- 2-port configuration for parallel bus master
- Channels can be assigned to either high-priority queue or low-priority queue. Within each queue, fixed or round-robin priorities can be serviced
- Built-in ECC generation and evaluation logic for internal RAM-storing channel transfer information
- Supports multiple interrupt outputs for mapping to multiple interrupt controllers in multicore systems
- 48 requests can be mapped to any 32 channels
- Supports LE endianness
- External ECC Gen/Eval block of MSS_DMA support ECC generation for data transactions, and parity for address, and control signals (following Cortex-R5F standard)
- 8 MPU regions
- Channel-chaining capability
- Hardware and software MSS_DMA requests

- 8-, 16-, 32-, or 64-bit transactions supported
- Multiple addressing modes for source and destination (fixed, increment, offset)
- Auto-initiation

2.6.4.3 MSS_DMA Request Map

Both instances of MSS_DMA have 64 lines of request and are connected to identical input triggers, as shown in Table 3-10. This allows the two DMAs to trigger different types of transfers for the same request.

Table 2-27. MSS_DMA Request Map

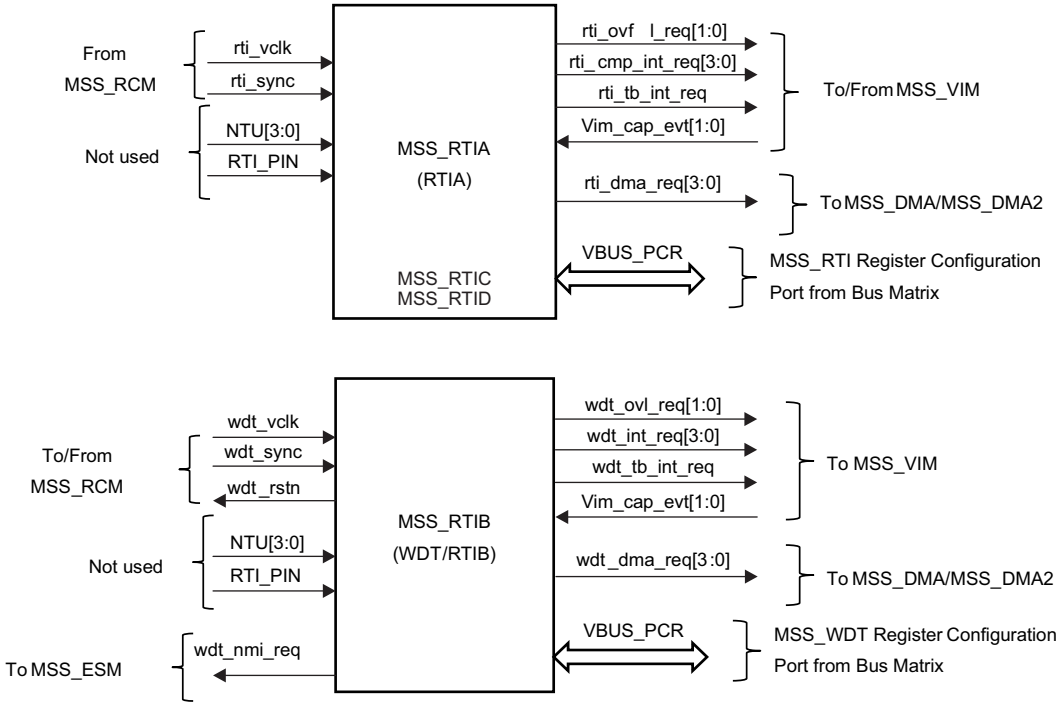
Module	DMA Request Sources	DMA Request
MSS_MIBSPIA	MSS_MIBSPIA Channel-1	DMAREQ[0]
MSS_MIBSPIA	MSS_MIBSPIA Channel-0	DMAREQ[1]
MSS_MIBSPIB	MSS_MIBSPIB	DMAREQ[2]
MSS_MIBSPIB	MSS_MIBSPIB	DMAREQ[3]
MSS_QSPI	MSS_QSPI DMA request	DMAREQ[4]
MSS_MIBSPIA	MSS_MIBSPIA Channel-3	DMAREQ[5]
MSS_DCAN	MSS_DCAN IF2	DMAREQ[6]
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) DMAREQ	DMAREQ[7]
MSS_DCAN	MSS_DCAN IF1	DMAREQ[8]
MSS_MIBSPIA	MSS_MIBSPIA Channel-5	DMAREQ[9]
MSS_I2C	MSS_I2C receive	DMAREQ[10]
MSS_I2C	MSS_I2C transmit	DMAREQ[11]
MSS_RTIA	MSS_RTIA DMAREQ0	DMAREQ[12]
MSS_RTIA	MSS_RTIA DMAREQ1	DMAREQ[13]
Reserved	Reserved	DMAREQ[14]
Reserved	Reserved	DMAREQ[15]
MSS_DCAN	MSS_DCAN IF3	DMAREQ[16]
MSS_MIBSPIA	MSS_MIBSPIA Channel-2	DMAREQ[17]
MSS_RTIA	MSS_RTIA DMAREQ2	DMAREQ[18]
MSS_RTIA	MSS_RTIA DMAREQ3	DMAREQ[19]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ0	DMAREQ[20]
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ1	DMAREQ[21]
MSS_MIBSPIA	MSS_MIBSPIA Channel-4	DMAREQ[22]
MSS_ETPWM3A	MSS_ETPWM3A DMAREQ	DMAREQ[23]
MSS_RTIB (WDT/RTIB)	WDT/RTIB DMAREQ2	DMAREQ[24]
MSS_RTIB (WDT/RTIB)	WDT/RTIB DMAREQ3	DMAREQ[25]
MSS_MCRC (CRC)	MSS_MCRC (CRC) DMAREQ0	DMAREQ[26]
MSS_MCRC (CRC)	MSS_MCRC (CRC) DMAREQ1	DMAREQ[27]
MSS_SCIB (UART2)	MSS_SCIB (UART2) receive	DMAREQ[28]
MSS_SCIB (UART2)	MSS_SCIB (UART2) transmit	DMAREQ[29]
MSS_SCIA (UART1)	MSS_SCIA (UART1) receive	DMAREQ[30]
MSS_SCIA (UART1)	MSS_SCIA (UART1) transmit	DMAREQ[31]
MSS_GIO	MSS_GIO-0	DMAREQ[32]
MSS_GIO	MSS_GIO-1	DMAREQ[33]
MSS_GIO	MSS_GIO-2	DMAREQ[34]
MSS_ETPWM1A	MSS_ETPWM1A DMAREQ	DMAREQ[35]
Reserved	Reserved	DMAREQ[36]
MSS_MIBSPIB	MSS_MIBSPIB Channel-2	DMAREQ[37]
MSS_MIBSPIB	MSS_MIBSPIB Channel-3	DMAREQ[38]
MSS_ETPWM1B	MSS_ETPWM1B DMAREQ	DMAREQ[39]

Table 2-27. MSS_DMA Request Map (continued)

Module	DMA Request Sources	DMA Request
MSS_ETPWM2A	MSS_ETPWM2A DMAREQ	DMAREQ[40]
MSS_ETPWM2B	MSS_ETPWM2B DMAREQ	DMAREQ[41]
MSS_MIBSPIB	MSS_MIBSPIB Channel-4	DMAREQ[42]
MSS_MIBSPIB	MSS_MIBSPIB Channel-5	DMAREQ[43]
Reserved	Reserved	DMAREQ[44]
MSS_ETPWM3B	MSS_ETPWM3B DMAREQ	DMAREQ[45]
MSS_GIO	MSS_GIO-14	DMAREQ[46]
MSS_GIO	MSS_GIO-15	DMAREQ[47]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-0	DMAREQ[48]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-1	DMAREQ[49]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-2	DMAREQ[50]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-3	DMAREQ[51]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-4	DMAREQ[52]
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-5	DMAREQ[53]
MSS_DTHE (Crypto/AES)	AES DMAREQ-0	DMAREQ[54]
MSS_DTHE (Crypto/AES)	AES DMAREQ-1	DMAREQ[55]
MSS_DTHE (Crypto/AES)	AES DMAREQ-2	DMAREQ[56]
MSS_DTHE (Crypto/AES)	AES DMAREQ-3	DMAREQ[57]
MSS_DTHE (Crypto/AES)	AES DMAREQ-4	DMAREQ[58]
MSS_DTHE (Crypto/AES)	AES DMAREQ-5	DMAREQ[59]
MSS_DTHE (Crypto/AES)	AES DMAREQ-6	DMAREQ[60]
MSS_DTHE (Crypto/AES)	AES DMAREQ-7	DMAREQ[61]
Reserved	Reserved	DMAREQ[62]
Reserved	Reserved	DMAREQ[63]

2.6.5 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB)

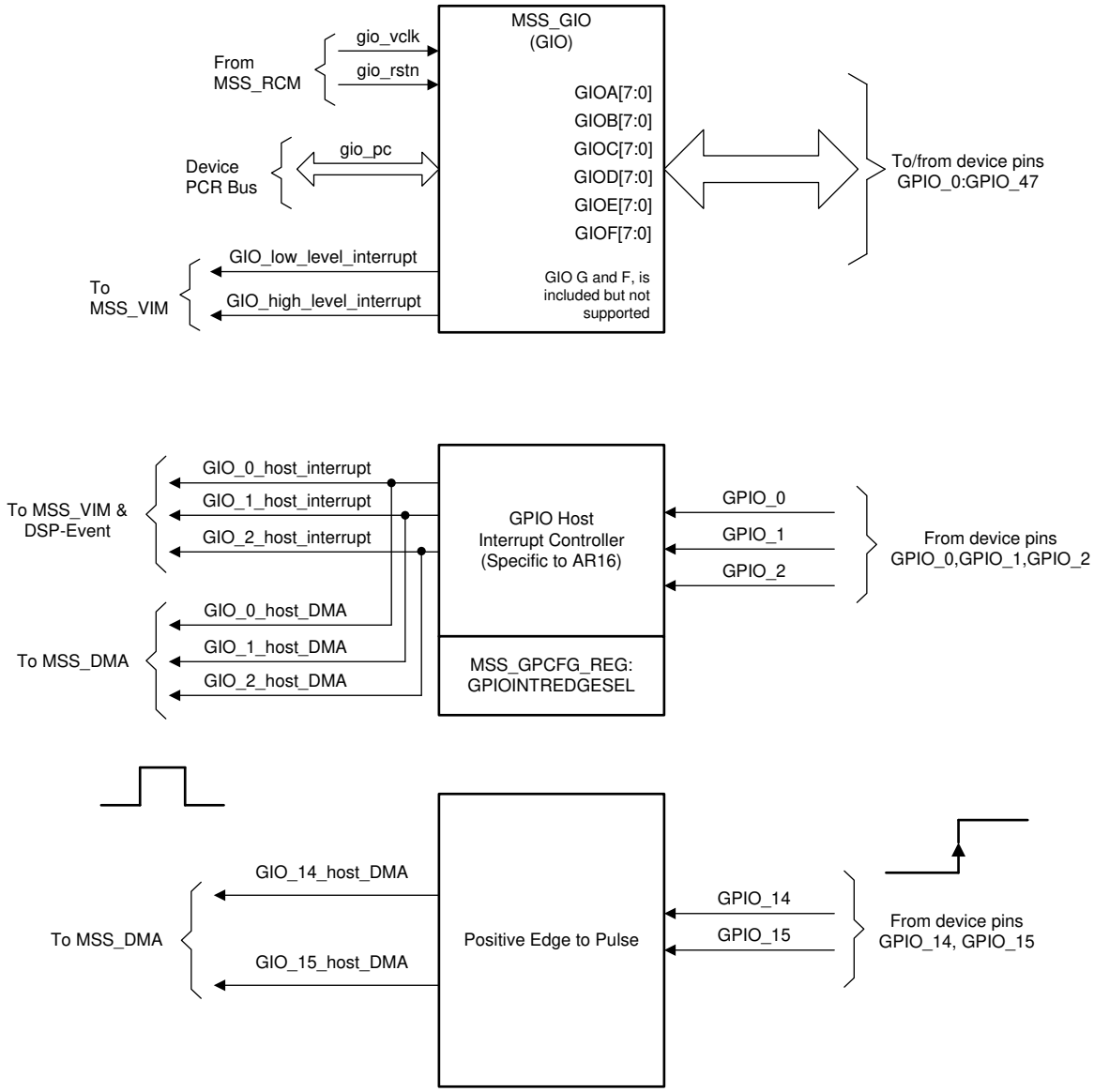
Figure 2-23. Integration of MSS_RTIA and MSS_RTIB, WDT Using the MSS_RTIB Module



2.6.6 General Purpose I/0 (MSS_GIO)

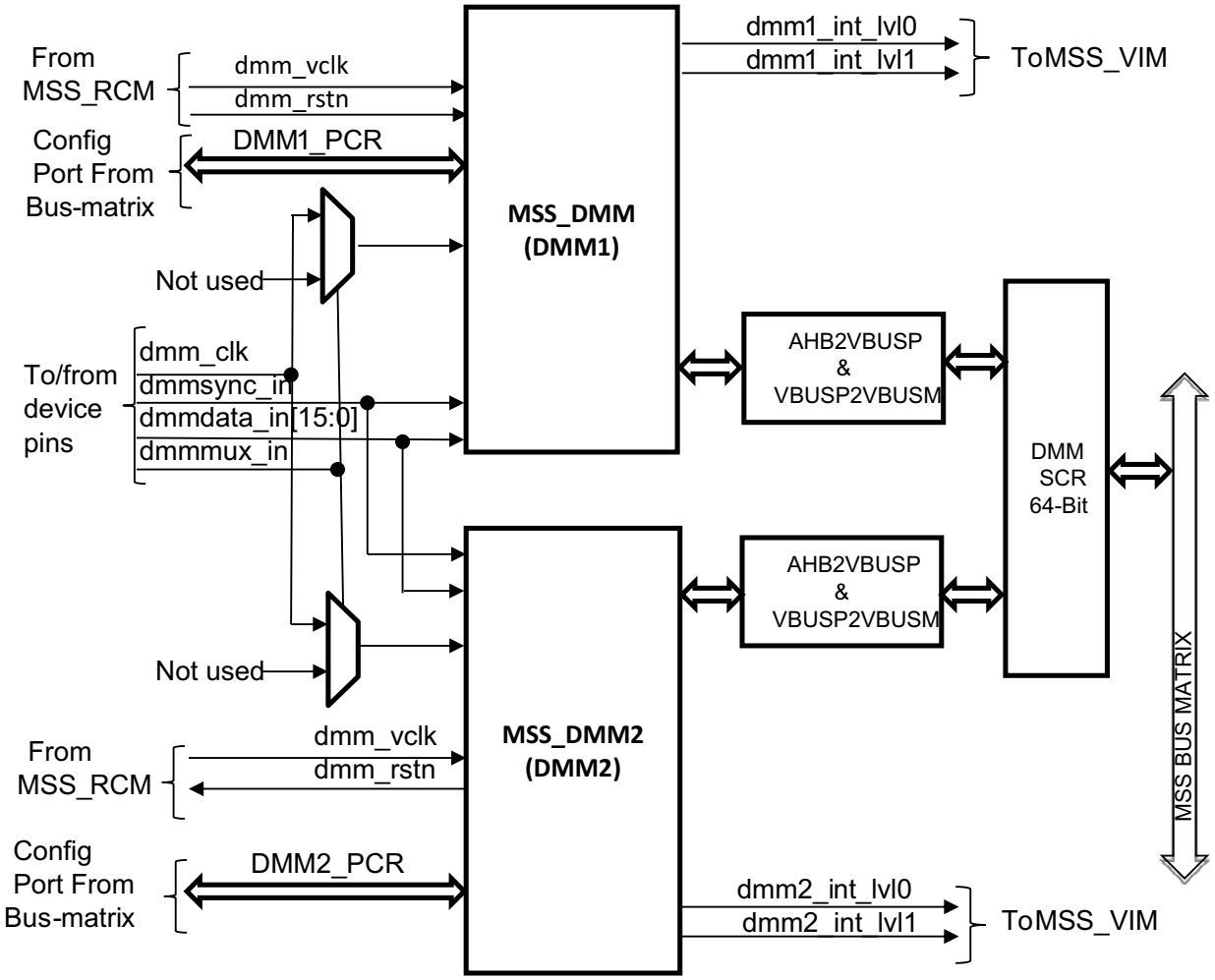
NOTE: Emulation mode and power-down mode (low-power mode) are not supported in the 18xx device.

Figure 2-24. Integration Block Diagram for MSS_GIO



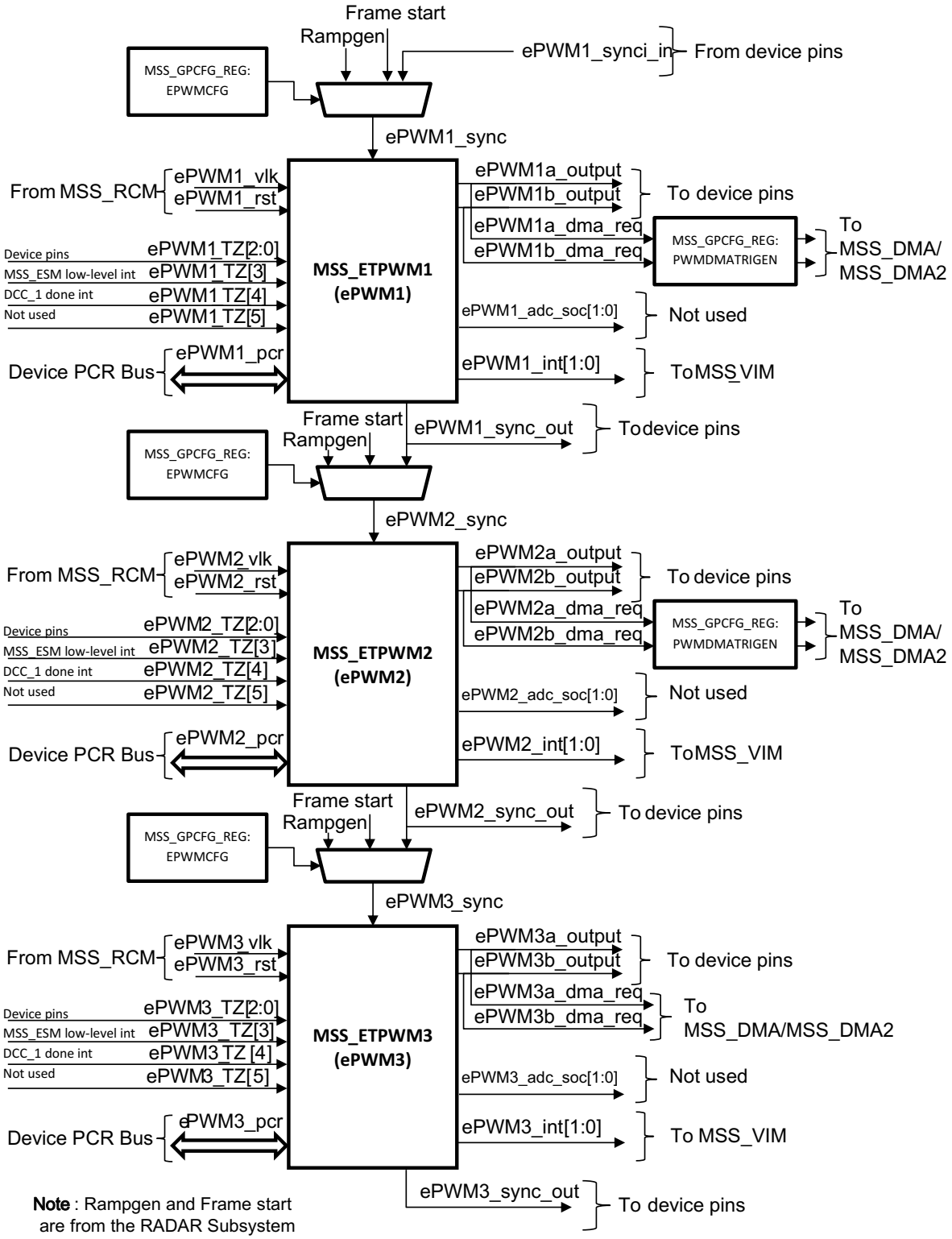
2.6.7 Data Modification Module (MSS_DMM)

Figure 2-25. MSS_DMM Integration



2.6.8 Enhanced Pulse Width Modulator (MSS_ETPWM)

Figure 2-26. Multiple MSS_ETPWM Modules



2.6.9 Vectored Interrupt Manager (MSS_VIM)

2.6.9.1 Interrupt Request Assignments

Table 2-28. Interrupt Request Assignments

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
MSS_ESM	MSS_ESM high-level interrupt(NMI)	0
Reserved	Reserved	1
MSS_RTIA	MSS_RTIA compare interrupt 0	2
MSS_RTIA	MSS_RTIA compare interrupt 1	3
MSS_RTIA	MSS_RTIA compare interrupt 2	4
MSS_RTIA	MSS_RTIA compare interrupt 3	5
MSS_RTIA	MSS_RTIA overflow interrupt 0	6
MSS_RTIA	MSS_RTIA overflow interrupt 1	7
MSS_RTIA	MSS_RTIA time-base	8
MSS_GIO	MSS_GIO high-level interrupt	9
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 0	10
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt1	11
MSS_MIBSPIA	MSS_MIBSPIA level 0 interrupt	12
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 2	13
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 3	14
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) overflow interrupt 0	15
MSS_DCAN	MSS_DCAN level 0 interrupt	16
MSS_MIBSPIB	MSS_MIBSPIB level 0 Interrupt	17
MSS_GIO host interrupt module	MSS_GIO GPIO_0_host_interrupt	18
MSS_MCRC (CRC)	MSS_MCRC (CRC) interrupt	19
MSS_ESM	MSS_ESM low-level interrupt	20
SYSTEM	Software-triggered interrupt 4	21
MSS Cortex R4F	MSS Cortex R4F interrupt PMU	22
MSS_GIO	MSS_GIO low-level interrupt	23
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) overflow interrupt 1	24
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) TB base interrupt	25
MSS_MIBSPIA	MSS_MIBSPIA level 0 interrupt	26
MSS_QSPI	MSS_QSPI interrupt	27
MSS_DMM	MSS_DMM S/W interrupt 38	28
MSS_DCAN	MSS_DCAN level 1 interrupt	29
MSS_MIBSPIB	MSS_MIBSPIB level 1 interrupt	30
MSS_DTHE (Crypto/SHA)	MSS_DTHE (Crypto/SHA) SHA -S interrupt	31
MSS_GIO host interrupt module	MSS_GIO GPIO_1_host_interrupt	32
MSS_DMA	MSS_DMA frame transfer complete interrupt	33
MSS_DMA	MSS_DMA last frame transfer start interrupt	34
Reserved	Reserved	35
MSS_DMM	MSS_DMM level -0 interrupt	36
MSS_DTHE (Crypto/SHA)	MSS_DTHE (Crypto/SHA) SHA -P interrupt	37
MSS_DTHE (Crypto/TRNG)	MSS_DTHE (Crypto/TRNG) TRNG interrupt	38

Table 2-28. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
MSS_DMA	MSS_DMA half-block transfer complete interrupt	39
MSS_DMA	MSS_DMA block transfer complete interrupt	40
MSS_DMA2	MSS_DMA2 frame block transfer complete interrupt	41
Reserved	Reserved	42
MSS_DMM	MSS_DMM level -1 interrupt	43
Reserved	Reserved	44
MSS_DMA2	MSS_DMA2 last frame complete interrupt	45
Reserved	Reserved	46
FPU	Floating point unit interrupt	47
MSS_GIO (GPIO host interrupt module)	MSS_GIO GPIO_2_host_interrupt	48
MSS_DMA2	MSS_DMA2 half-block transfer complete interrupt	49
MSS_DMA2	MSS_DMA2 block transfer complete interrupt	50
MSS_DMA2	MSS_DMA2 bus error interrupt	51
System	DSS to MSS software-triggered by register DSS_REG2:MSSSWIRQ:MSSSWIRQ1	52
MSS_DTHE (Crypto/PKA)	MSS_DTHE (Crypto/PKA) PKA module interrupt	53
MSS_DTHE (Crypto/AES)	MSS_DTHE (Crypto/AES) AES-S module interrupt	54
Reserved	Reserved	55
MSS_DTHE (Crypto/AES)	MSS_DTHE (Crypto/AES) AES-P module interrupt	56
MSS_DMM2	MSS_DMM2 level -0 interrupt	57
MSS_DMM2	MSS_DMM2 level -1 interrupt	58
Mailbox	DSS to MSS mailbox full interrupt	59
Mailbox	DSS to MSS mailbox empty interrupt	60
System	DSS to MSS software-triggered by register DSS_REG2:MSSSWIRQ:MSSSWIRQ2	61
MSS_DEBUGSS (Debug subsystem)	MSS_DEBUGSS (Debug subsystem) interrupt	62
DSPSS-MSS_STC	GEM MSS_STC done interrupt	63
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 0 interrupt	64
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 0 interrupt	65
MSS_I2C	MSS_I2C interrupt	66
MSS_DMM	MSS_DMM interrupt 34	67
MSS_DMM	MSS_DMM interrupt 35	68
MSS_DMM	MSS_DMM interrupt 36	69
MSS_DMA	MSS_DMA bus error interrupt	70
MSS_DMM/Radar subsystem	MSS_DMM interrupt 30 or Radar subsystem logical Frame Start	71
Reserved	Reserved	72
MSS_DMM	MSS_DMM interrupt 33	73
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 1 interrupt	74
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 1 interrupt	75
SYSTEM	Software-triggered interrupt 0	76

Table 2-28. Interrupt Request Assignments (continued)

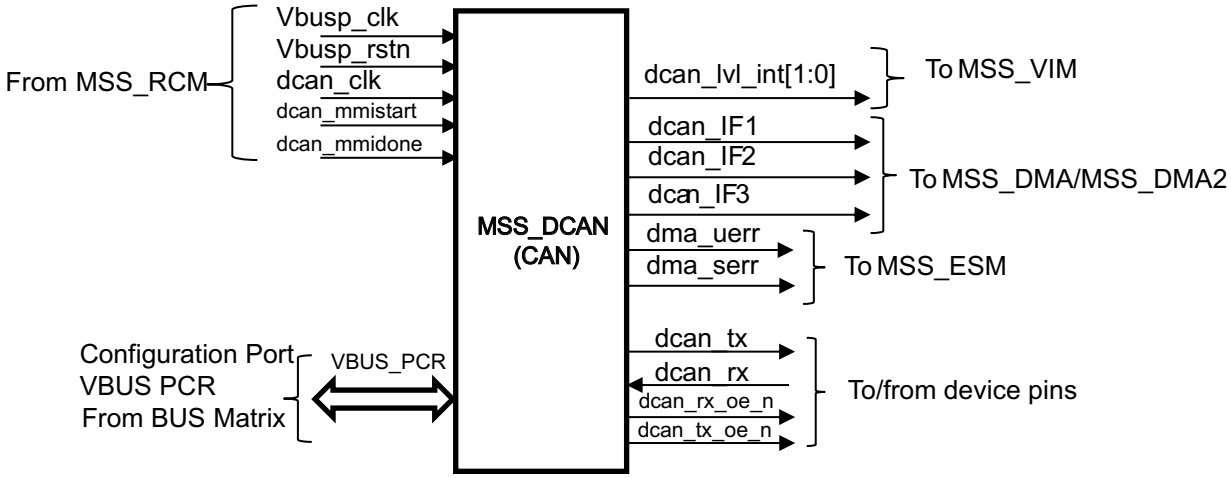
Module	VIM Interrupt Sources	Default VIM Interrupt Channel
SYSTEM	Software-triggered interrupt 1	77
SYSTEM	Software-triggered interrupt 2	78
SYSTEM	Software-triggered interrupt 3	79
Reserved	Reserved	80
Reserved	Reserved	81
MSS_DCCA	MSS_DCCA (dual clock compare) module1-done interrupt	82
MSS_DCCB	MSS_DCCB (dual clock compare) module2-done interrupt	83
SYSTEM	Software-triggered interrupt 5	84
MSS_PBIST	MSS_PBIST interrupt	85
MSS_DMM/DSS	GEM IRQ-7/MSS_DMM interrupt 32	86
Reserved	Reserved	87
Reserved	Reserved	88
Reserved	Reserved	89
Reserved	Reserved	90
Reserved	Reserved	91
Reserved	Reserved	92
Reserved	Reserved	93
Reserved	Reserved	94
MAILBOX	RADARSS to MSS mailbox interrupt	95
MAILBOX	RADARSS mailbox read complete interrupt sent from RADARSS to MSS	96
RADARSS	ADC valid fall interrupt	97
MSS_DMM/RADARSS	MSS_DMM interrupt 29/frame start interrupt/	98
RADARSS	Chirp start interrupt	99
RADARSS	Chirp end Interrupt	100
RADARSS	Frame end Interrupt	101
Reserved	Reserved	102
Reserved	Reserved	103
MSS_ETPWM1	ePWM1 interrupt-1	104
RADARSS-MSS_STC	MSS_STC done Interrupt	105
RadarSS	All RadarSS interrupts combined	106
MSS_ETPWM1	ePWM1 interrupt-2	107
MSS_ETPWM2	ePWM2 interrupt-1	108
MSS_ETPWM2	ePWM2 interrupt-2	109
MSS_ETPWM3	ePWM3 interrupt-1	110
MSS_ETPWM3	ePWM3 interrupt-2	111
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) interrupt	112
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) error interrupt	113
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) interrupt	114
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) error interrupt	115
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) interrupt	116
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) error interrupt	117
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) interrupt	118

Table 2-28. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel
Reserved	Reserved	119
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) error interrupt	120
MSS_DMM	MSS_DMM interrupt 37	121
Reserved	Reserved	122
DSS_ADCBUF/MSS_DMM	Chirp available interrupt/MSS_DMM interrupt 31	123
MSS_PBIST	MSS_PBIST: Gem MSS_STC done	124
DSS_HW_ACC	DSS_HW_ACC FFT accelerator -param done interrupt	125
DSS_HW_ACC	DSS_HW_ACC FFT accelerator - done interrupt	126
DSS_HW_ACC	DSS_HW_ACC FFT accelerator –access error interrupt	127

2.6.10 Controller Area Network (MSS_DCAN)

Figure 2-27. Integration Block Diagram for MSS_DCAN Module



2.6.11 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPI)

Figure 2-28. MSS_MIBSPIA Integration

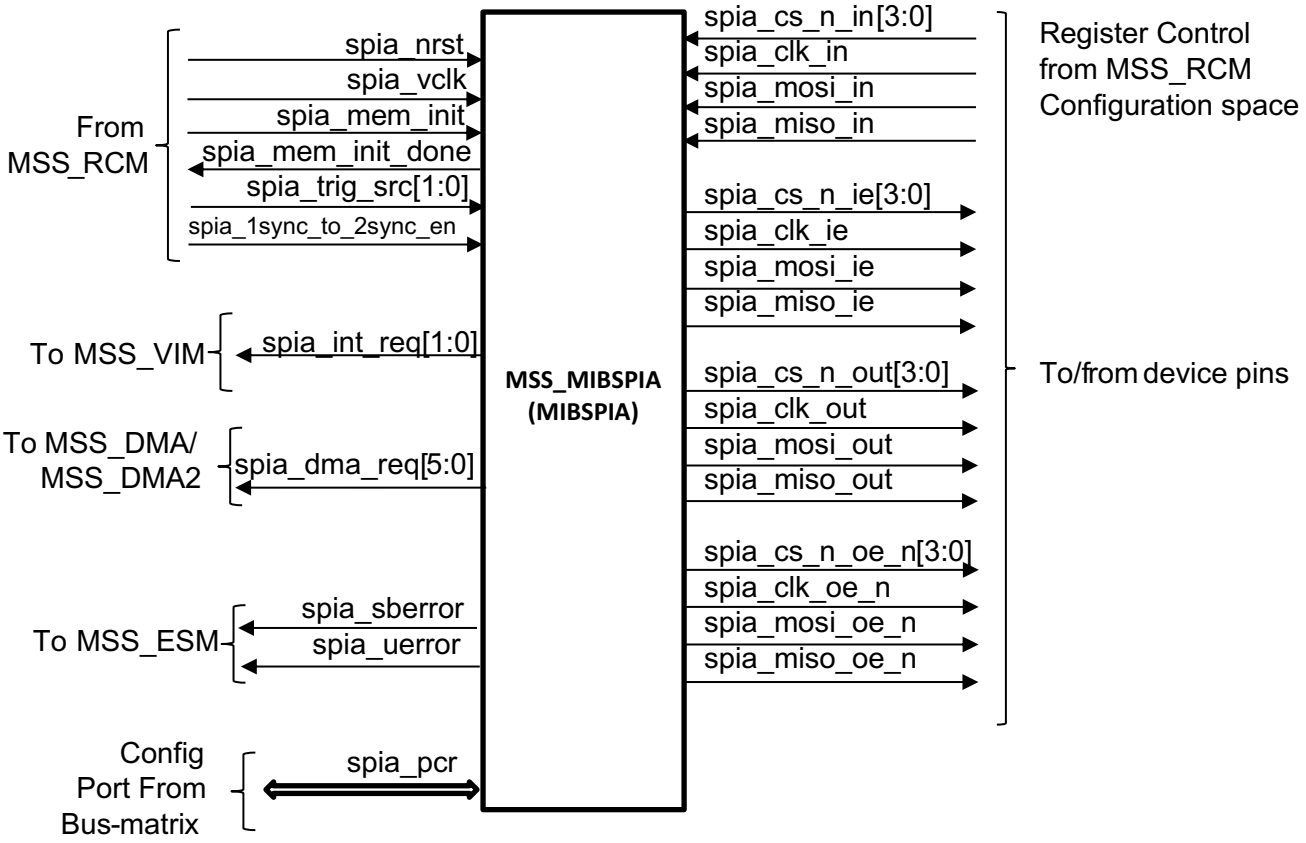
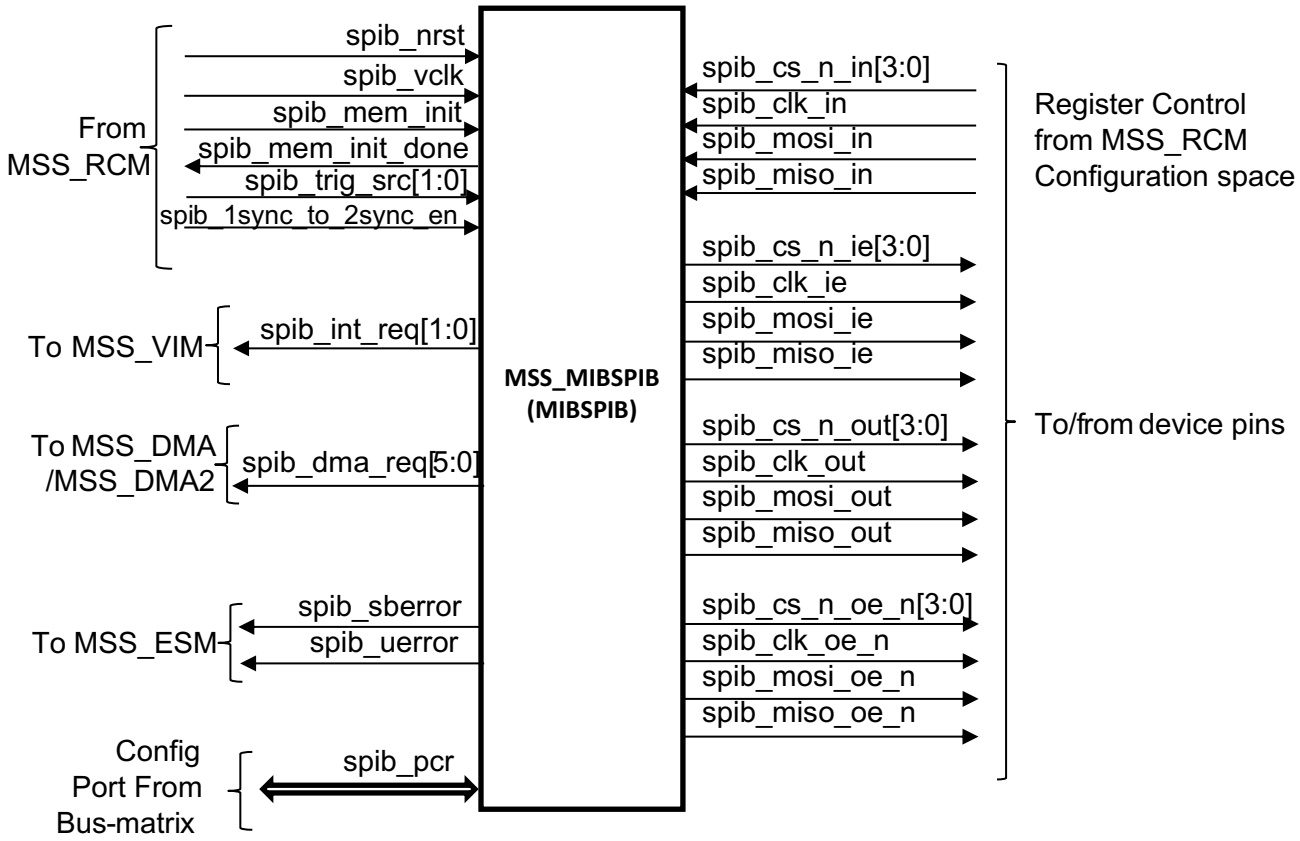


Figure 2-29. MSS_MIBSPIB Integration



2.6.12 Quad Serial Peripheral Interface (MSS_QSPI)

The MSS_QSPI module of the 18xx only supports one CS pin, (qspi1_cs).

2.6.13 Enhanced Direct Memory Access (EDMA)

2.6.13.1 EDMA Controller Integration

The 18xx device has two EDMA channel controllers (DSS_TPCC0 and DSS_TPCC1) on the device:

- DSS_TPCC0 (EDMA TPCC0) has two transfer controllers: DSS_TPTC0 (EDMA TPTC0) and DSS_TPTC1 (EDMA TPTC1)
- DSS_TPCC1 (EDMA TPCC1) has two transfer controllers: DSS_TPTC2 (EDMA TPTC2) and DSS_TPTC3 (EDMA TPTC3)

NOTE: The 18xx device does not support the region interrupt feature of the EDMA peripheral. Only the global interrupt feature of the EDMA module is supported.

Table 2-29. DSS_TPCC Configuration

	DSS_TPCC0 (EDMA TPCC0)	DSS_TPCC1 (EDMA TPCC1)
Number of MSS_DMA channels	64	64
Number of PaRAM entires	128	256
Number of QDMA channels	8	8

Table 2-29. DSS_TPCC Configuration (continued)

	DSS_TPCC0 (EDMA TPCC0)	DSS_TPCC1 (EDMA TPCC1)
Number of event queues	2	2
Memory protection existence	No	No
Channel mapping	No	No
Number of TCs (transfer controllers)	2	2

Table 2-30. DSS_TPTC Configuration

	DSS_TPTC[0-1]	DSS_TPTC[2-3]
FIFO size	512 bytes	128 bytes
TR pipe depth	2	2
Bus width	16 bytes	16 bytes

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 3-13 and Figure 3-14 show the EDMA controller integration.

Figure 2-30. EDMA Controller Integration (1 of 2)

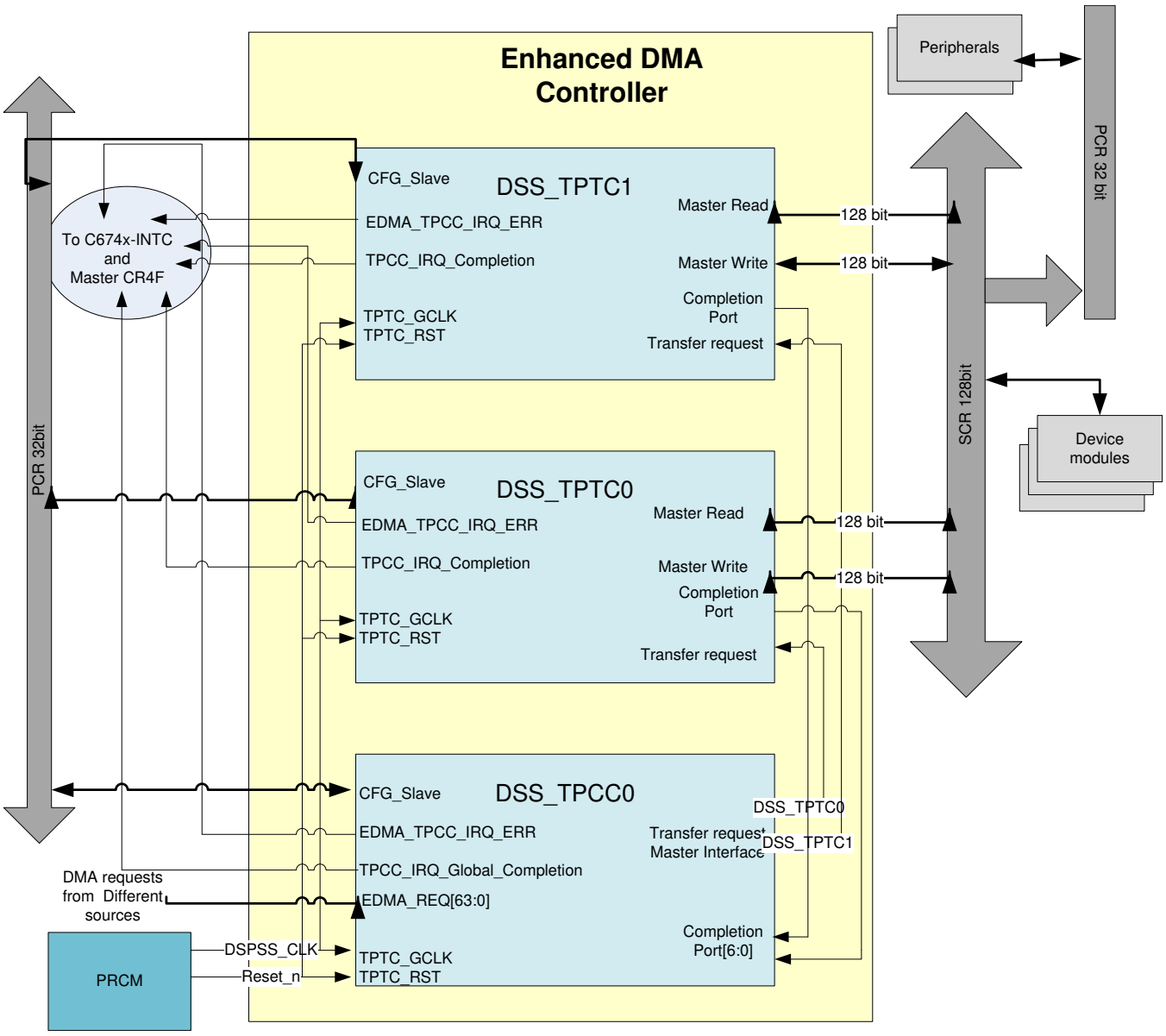
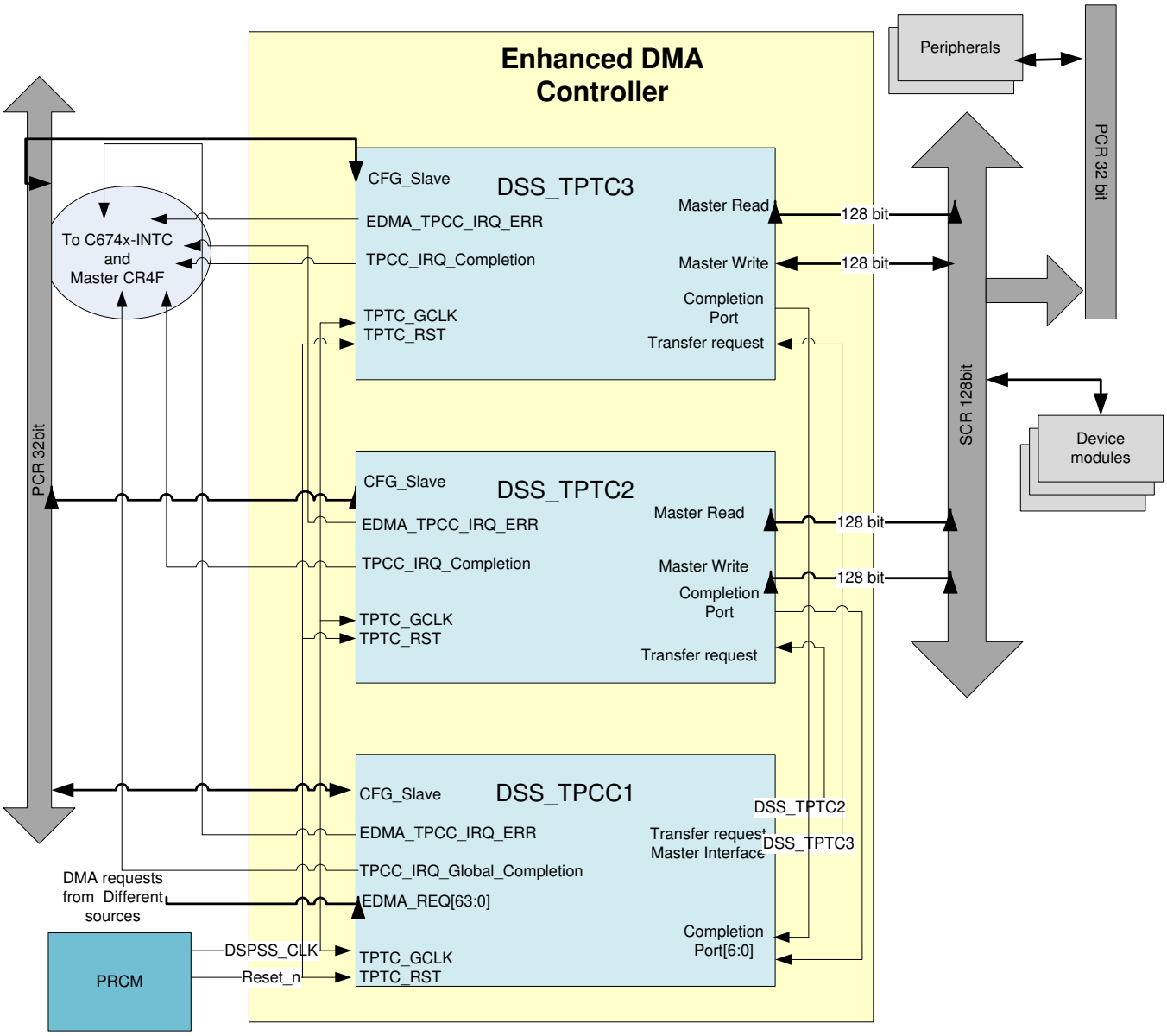


Figure 2-31. EDMA Controller Integration (2 of 2)



2.6.13.2 EDMA Request Map

Table 2-31. EDMA Request Map

Request Number	Hardware Event
DSS_TPCC0 (EDMA TPCC0) DMA	
0	DSS_CBUFF_DMA_REQ_0
1	DSS_CBUFF_DMA_REQ_1
2	DSS_CBUFF_DMA_REQ_2
3	DSS_CBUFF_DMA_REQ_3
4	DSS_CBUFF_DMA_REQ_4
5	DSS_CBUFF_DMA_REQ_5
6	DSS_CBUFF_DMA_REQ_6
7	RESERVED

Table 2-31. EDMA Request Map (continued)

Request Number	Hardware Event
8	Frame Start/DSS_DMMSWINT9/DSS_DMMSWINT39
9	Chirp Available/DSS_DMMSWINT11/DSS_DMMSWINT43
10	RESERVED
11	RESERVED
12	RESERVED
13	RESERVED
14	RESERVED
15	RESERVED
16	RESERVED
17	DSS_FFT_ACC_CHANNEL_TRIGGER_0
18	DSS_FFT_ACC_CHANNEL_TRIGGER_1
19	DSS_FFT_ACC_CHANNEL_TRIGGER_2
20	DSS_FFT_ACC_CHANNEL_TRIGGER_3
21	DSS_FFT_ACC_CHANNEL_TRIGGER_4
22	DSS_FFT_ACC_CHANNEL_TRIGGER_5
23	DSS_FFT_ACC_CHANNEL_TRIGGER_6
24	DSS_FFT_ACC_CHANNEL_TRIGGER_7
25	DSS_FFT_ACC_CHANNEL_TRIGGER_8
26	DSS_FFT_ACC_CHANNEL_TRIGGER_9
27	DSS_FFT_ACC_CHANNEL_TRIGGER_10
28	DSS_FFT_ACC_CHANNEL_TRIGGER_11
29	DSS_FFT_ACC_CHANNEL_TRIGGER_12
30	DSS_FFT_ACC_CHANNEL_TRIGGER_13
31	DSS_FFT_ACC_CHANNEL_TRIGGER_14
32	DSS_FFT_ACC_CHANNEL_TRIGGER_15
33	DSS_MCRC_DMA_REQ_0
34	DSS_MCRC_DMA_REQ_1
35	FRC_EVENT_GEN_0
36	FRC_EVENT_GEN_1
37	FRC_EVENT_GEN_2
38	FRC_EVENT_GEN_3
39	RESERVED
40	LOGICAL_FRAME_START/DSS_DMMSWINT10/DSS_DMMSWINT40
41	ADC_DATA_VALID_FALL/DSS_DMMSWINT12/DSS_DMMSWINT44
42	UART_DMA_REQ_0
43	UART_DMA_REQ_1
44	DMMSW_INT_13
45	DMMSW_INT_14
46	DMMSW_INT_15
47	DMMSW_INT_16
48	DMMSW_INT_17
49	GPIO_0_host_interrupt
50	GPIO_1_host_interrupt
51	GPIO_2_host_interrupt
52	RTI1_DMA_REQ_0
53	RTI1_DMA_REQ_1
54	RTI1_DMA_REQ_2

Table 2-31. EDMA Request Map (continued)

Request Number	Hardware Event
55	RTI1_DMA_REQ_3
56	RTI2_DMA_REQ_0
57	RTI2_DMA_REQ_1
58	RTI2_DMA_REQ_2
59	RTI2_DMA_REQ_3
60	RESERVED
61	RESERVED
62	RESERVED
63	DMMSW_INT_18
DSS_TPCC1 (EDMA TPCC1) MSS_DMA	
0	DSS_CBUFF_DMA_REQ_0
1	DSS_CBUFF_DMA_REQ_1
2	DSS_CBUFF_DMA_REQ_2
3	DSS_CBUFF_DMA_REQ_3
4	DSS_CBUFF_DMA_REQ_4
5	DSS_CBUFF_DMA_REQ_5
6	DSS_CBUFF_DMA_REQ_6
7	RESERVED
8	FRAME_START/DSS_DMMSWINT19/DSS_DMMSWINT39
9	CHIRP_AVIALABLE/DSS_DMMSWINT21/DSS_DMMSWINT43
10	RESERVED
11	RESERVED
12	RESERVED
13	RESERVED
14	RESERVED
15	RESERVED
16	RESERVED
17	DSS_FFT_ACC_CHANNEL_TRIGGER_0
18	DSS_FFT_ACC_CHANNEL_TRIGGER_1
19	DSS_FFT_ACC_CHANNEL_TRIGGER_2
20	DSS_FFT_ACC_CHANNEL_TRIGGER_3
21	DSS_FFT_ACC_CHANNEL_TRIGGER_4
22	DSS_FFT_ACC_CHANNEL_TRIGGER_5
23	DSS_FFT_ACC_CHANNEL_TRIGGER_6
24	DSS_FFT_ACC_CHANNEL_TRIGGER_7
25	DSS_FFT_ACC_CHANNEL_TRIGGER_8
26	DSS_FFT_ACC_CHANNEL_TRIGGER_9
27	DSS_FFT_ACC_CHANNEL_TRIGGER_10
28	DSS_FFT_ACC_CHANNEL_TRIGGER_11
29	DSS_FFT_ACC_CHANNEL_TRIGGER_12
30	DSS_FFT_ACC_CHANNEL_TRIGGER_13
31	DSS_FFT_ACC_CHANNEL_TRIGGER_14
32	DSS_FFT_ACC_CHANNEL_TRIGGER_15
18	RESERVED
19	RESERVED
20	RESERVED
21	RESERVED

Table 2-31. EDMA Request Map (continued)

Request Number	Hardware Event
22	RESERVED
23	RESERVED
24	RESERVED
25	RESERVED
26	RESERVED
27	RESERVED
28	RESERVED
29	RESERVED
30	RESERVED
31	RESERVED
32	RESERVED
33	DSS_MCRC_DMA_REQ_0
34	DSS_MCRC_DMA_REQ_1
35	FRC_EVENT_GEN_0
36	FRC_EVENT_GEN_1
37	FRC_EVENT_GEN_2
38	FRC_EVENT_GEN_3
39	RESERVED
40	LOGICAL_FRAME_START/DSS_DMMSWINT20/DSS_DMMSWINT40
41	ADC_DATA_VALID_FALL/DSS_DMMSWINT22/DSS_DMMSWINT44
42	UART_DMA_REQ_0
43	UART_DMA_REQ_1
44	DMMSW_INT_23
45	DMMSW_INT_24
46	DMMSW_INT_25
47	DMMSW_INT_26
48	DMMSW_INT_27
49	GPIO_0_host_interrupt
50	GPIO_1_host_interrupt
51	GPIO_2_host_interrupt
52	RTI1_DMA_REQ_0
53	RTI1_DMA_REQ_1
54	RTI1_DMA_REQ_2
55	RTI1_DMA_REQ_3
56	RTI2_DMA_REQ_0
57	RTI2_DMA_REQ_1
58	RTI2_DMA_REQ_2
59	RTI2_DMA_REQ_3
60	RESERVED
61	RESERVED
62	RESERVED
63	DMMSW_INT_28

2.6.14 Error Signaling Module (MSS_ESM/DSS_ESM)

The 18xx device has two instances of the Error Signaling Module (MSS_ESM/DSS_ESM), shown in Figure 3-15.

Figure 2-32. 18xx MSS_ESM/DSS_ESM Integration Diagram

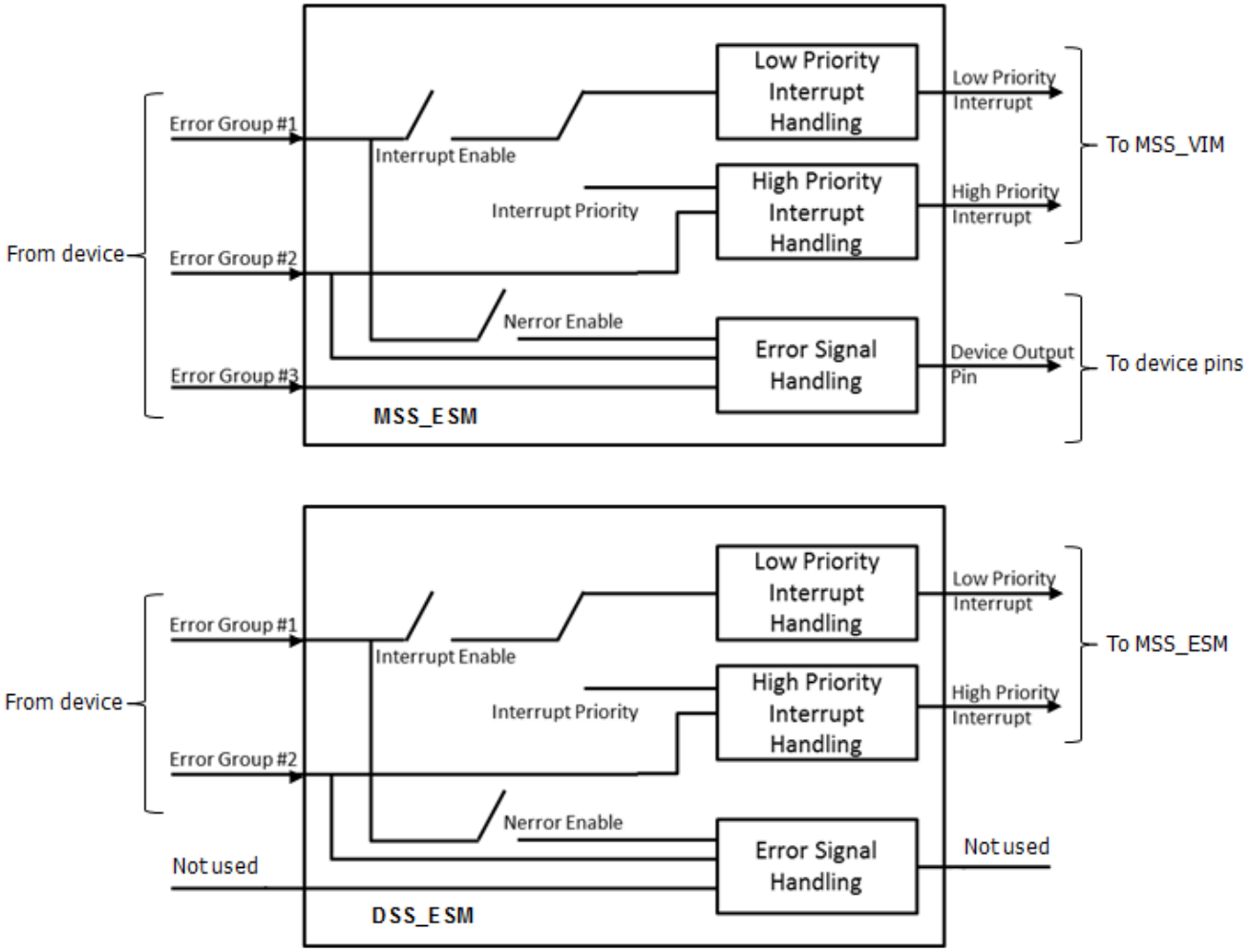


Table 3-15 shows the mapping on the input error inputs to the ESM module from various error sources available for hardware diagnostics within the device.

Table 2-32. MSS_ESM Mapping

MSS_ESM Group 1		Channel Type	Description
63	ANA_LIMP_MODE	Error Signal	Error signal at device boot-up, if the CLK monitor finds the REF CLK to be outside the permissible range of frequency
62	MSS_DCCB_ERR	Error Signal	MSS_DCCB frequency comparison error
61	MAILBOX_BSS2MSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2MSS
60	MAILBOX_BSS2MSS _REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_BSS2MSS
59	MAILBOX_MSS2BSS _FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2BSS
58	MAILBOX_MSS2BSS _REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2BSS
57	BSS_CRITICAL_ERR	Error Signal	BSS critical Error Indication. Mask control to be configured in IRQ_CTL for individual error signals.
56	Reserved	Reserved	Reserved

Table 2-32. MSS_ESM Mapping (continued)

MSS_ESM Group 1		Channel Type	Description
55	CLOCK_SUPPLY_ERR	Error Signal	Clock and Supply Errors from Analog. Mask control to be configured in IRQ_CTL for individual error signals
54	Reserved	Reserved	Reserved
53	MAILBOX_DSS2MSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2MSS
52	MAILBOX_DSS2MSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_DSS2MSS
51	Reserved	Reserved	Reserved
50	Reserved	Reserved	Reserved
49	MSS_MIBSPIB_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MIBSPIB multi-buffer (RXRAM/TXRAM)
48	MSS_MCRC_ERR	Error Signal	MSS_MCRC Comparison Error
47	Reserved	Reserved	Reserved
46	Reserved	Reserved	Reserved
45	MSS_MIBSPIB_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MIBSPIB multi-buffer (RXRAM/TXRAM)
44	Reserved	Reserved	Reserved
43	MAILBOX_MSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2DSS
42	MAILBOX_MSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2DSS
41	DSS_ESM_GP1_ERR	Error Signal	DSS_ESM Low priority Interrupt
40:39	Reserved	Reserved	Reserved
38	DSS_CBUFF_SAFETY_ERR	Error Signal	CHIRP ERROR or CRC ERROR from DSS_CBUFF
37	DSS_ESM_GP2_ERR	Error Signal	DSS_ESM High priority Interrupt
36	DSS_TPTC1_WR_MPU_ERR	Error Signal	DSS_TPTC1 write port MPU error
35	DSS_TPTC1_RD_MPU_ERR	Error Signal	DSS_TPTC1 read port MPU error
34	HVMODE_ERR	Error Signal	Error indication from IO Supply (Supply detector for dual-voltage IOs)
33	MSS_DCAN_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_DCAN Message RAM (FRAM/SRAM)
32	MSS_TCMA_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for Cortex R4F MSS_TCMA
31	Reserved	Reserved	Reserved
30	MSS_DCCA_ERR	Error Signal	MSS_DCCA frequency comparison error
29	DSS_TPTC0_WR_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
28	MSS_TCMB1_REPAIR_ERR	Alert Signal	Single-bit error/reserved indication for MSS_TCMB1
27	MSS_STC_ERR	Error Signal	MSS_STC Error indication for MSS Cortex R4F
26	MSS_TCMB0_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_TCMB0
25	MSS_MIBSPIA_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MIBSPIA multi-buffer (RXRAM/TXRAM)
24	MSS_DMA2_MEM_PARITY_ERR	Error Signal	Parity Error for MSS_DMA2 memory
23	MSS_DMA2_MPU_ERR	Error Signal	Error indication from MPU of MSS_DMA2
22	FRC_COMPARE_ERR	Error Signal	Lockstep comparison error from Free running Counter (FRC) in BSS
21	MSS_DCAN_RAM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_DCAN Message Memory (FRAM/SRAM)
20	MSS_VIM_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_VIM_RAM
19	Reserved	Reserved	Reserved
18	DSS_TPTC0_RD_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
17	MSS_MIBSPIA_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MIBSPIA (RXRAM/TXRAM)
16	MSS_SECURE_RAM_FATAL_ERR	Error Signal	Multi-bit uncorrectable error indication for MSS_DTHE/SECURE_RAM
15	MSS_VIM_RAM_FATAL_ERR	Error Signal	Multi-bit uncorrectable error indication for MSS_VIM_RAM

Table 2-32. MSS_ESM Mapping (continued)

MSS_ESM Group 1		Channel Type	Description
14	MSS_SECURE_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_DTHE/SECURE_RAM
13	Reserved	Reserved	Reserved
12	MAILBOX_BSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2DSS
11	MAILBOX_BSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_BSS2DSS
10	MAILBOX_DSS2BSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2BSS
9	DSS_CBUFF_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_CBUFF FIFO
8	DSS_CBUFF_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_CBUFF FIFO
7	DSS_TPCC_PARITY_ERR	Error Signal	Parity error from DSS_TPCC (EDMA Channel Controller)
6	NU	Reserved	Reserved
5	MAILBOX_DSS2BSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_DSS2BSS
4	MSS_CCCB_ERR	Error Signal	MSS_CCCB(Clock compare core) frequency comparison error
3	MSS_DMA_MEM_PARITY_ERR	Error Signal	Parity Error for DMA1 memory
2	MSS_DMA_MPU_ERR	Error Signal	Error indication from MPU of MSS _DMA
1	MSS_CCCA_ERR	Error Signal	MSS_CCCA(Clock compare core) frequency comparison error
0	NERROR_PAD_IN	Error Signal	Nerror from PAD looped in
MSS_ESM Group 2			
31:26	NU		
25	Reserved	Reserved	Reserved
24	MSS_RTIB_NMI	Error Signal	Watchdog Non-mask able interrupt
23:17	NU		
16	MSS_CR4F_LIVELOCK_ERR	Error Signal	Cortex R4F Live lock error
15:9	NU		
8	MSS_TCMB1_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB1
7	NU		
6	MSS_TCMB0_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB0
5	NU		
4	MSS_TCMA_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMA
3	NU		
2	Reserved	Reserved	Reserved
1:0	NU		
MSS_ESM Group 3			
31-8	NU		
7	MSS_TCMA_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMA
6	NU		
5	MSS_TCMB1_FATAL_ERR	Error Signal	Multi -bit error indication for MSS_TCMB1
4	NU		
3	MSS_TCMB0_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMB0
2	NU		
1	EFC_AUTOLOAD_ERR	Error Signal	Efuse Auto-load error
0	NU		

Table 2-33. DSS_ESM Mapping

DSS_ESM Group 1		Channel Type	Description
63:57	Reserved	Reserved	Reserved
56	DSS_DSP_L1P_PARITY_ERR	Error Signal	DSS_DSP_L1P Parity Error
55	MAILBOX_MSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2DSS
54	MAILBOX_DSS2MSS_REPAIR_ERR	Error Signal	Single-bit error/repair indication from MAILBOX_DSS2MSS
53	MAILBOX_DSS2MSS_FATAL_ERR	Alert Signal	Multi-bit error indication from MAILBOX_DSS2MSS
52	MAILBOX_MSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2DSS
51	MAILBOX_BSS2DSS_REPAIR_ERR	Alert Signal	Single-bit repair indication from MAILBOX_BSS2DSS
50	MAILBOX_DSS2BSS_REPAIR_ERR	Alert Signal	Single-bit repair indication from MAILBOX_DSS2BSS
49	MAILBOX_BSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2DSS
48	MAILBOX_DSS2BSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2BSS
47:32	Reserved	Reserved	Reserved
31	DSS_CFG_MSTID_MPU_ERR	Error Signal	Error from Master ID based MPU on the DSS_CFG_MSTID_MPU Configuration address space
30:29	Reserved	Reserved	Reserved
28	DSS_DATA_TXFR_RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_DATA_TXFR_RAM
27	DSS_DATA_TXFR_RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_DATA_TXFR_RAM
26	DSS_STC_ERR	Error Signal	Error from DSS_STC (Self-test Controller)
25	DSS_DSP_L2_UMAP_ECC_REPAIR_ERR	Alert Signal	Single bit repair indication for DSS_DSP_L2_UMAP0 or DSS_DSP_L2_UMAP1
24	DSS_HSRAM1_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_HSRAM1
23	DSS_HSRAM1_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_HSRAM1
22:19	Reserved	Reserved	Reserved
18	DSS_ADCBUF_PONG_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_ADCBUF Pong Memory
17	DSS_ADCBUF_PONG_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_ADCBUF Pong Memory
16	DSS_ADCBUF_PING_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_ADCBUF Ping Memory
15	DSS_ADCBUF_PING_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_ADCBUF Ping Memory
14	DSS_TPTC3_WR_MPU_ERR	Error Signal	DSS_TPTC3 write port MPU error
13	DSS_TPTC3_RD_MPU_ERR	Error Signal	DSS_TPTC3 read port MPU error
12	DSS_TPTC2_WR_MPU_ERR	Error Signal	DSS_TPTC2 write port MPU error
11	DSS_TPTC2_RD_MPU_ERR	Error Signal	DSS_TPTC2 read port MPU error
10	DSS_TPCC1_PARITY_ERR	Error Signal	Parity error from DSS_TPCC1 (EDMA Channel Controller)
9	DSS_CBUFF_SAFETY_ERR	Error Signal	CHIRP ERROR or CRC ERROR from DSS_CBUFF
8	DSS_TPTC1_WR_MPU_ERR	Error Signal	DSS_TPTC1 write port MPU error
7	DSS_TPTC1_RD_MPU_ERR	Error Signal	DSS_TPTC1 read port MPU error
6	DSS_TPTC0_WR_MPU_ERR	Error Signal	DSS_TPTC0 write port MPU error
5	DSS_TPTC0_RD_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
4	DSS_CBUFF_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_CBUFF FIFO
3	DSS_CBUFF_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_CBUFF FIFO
2	DSS_TPCC_PARITY_ERR	Error Signal	Parity error from DSS_TPCC (EDMA Channel Controller)
1	DSS_L3RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_L3RAM Memory
0	DSS_L3RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_L3RAM Memory
DSS_ESM Group 2			
31:6	Reserved	Reserved	Reserved
5	DSS_DSP_L2_UMAP_ECC_FATAL_ERR	Error Signal	Multi-bit error indication for DSS_DSP_L2_UMAP0 or DSS_DSP_L2_UMAP1
4	DSP_PBIST_ERR	Error Signal	DSP_PBIST (Memory Test) Fail Error

Table 2-33. DSS_ESM Mapping (continued)

DSS_ESM Group 1		Channel Type	Description
3	DSS_STC_ERR	Error Signal	Error from DSS_STC (Self-test Controller)
2	DSS_RTI_NMI	Error Signal	NMI from DSS_RTI (Watchdog)
1	DSS_DSP_L2_UMAP1_PARITY_ERR	Error Signal	Error from byte level parity comparison logic for DSS_DSP_L2_UMAP1
0	DSS_DSP_L2_UMAP0_PARITY_ERR	Error Signal	Error from byte level parity comparison logic for DSS_DSP_L2_UMAP0

2.6.15 High-Speed Interface (HSI)

Table 3-17 lists the high-speed interfaces available for the 18xx device.

Table 2-34. 18xx High-Speed Interfaces

CSI2	N/A
LVDS	2 lanes

2.6.16 Handshake RAM (DSS_HSRAM1)

The 18xx device has 32KB memory for HSRAM in the DSP subsystem.

68xx/64xx

Topic Page

3.1 68xx/64xx Introduction 247
3.2 Memory Map 251
3.3 68xx/64xx Integration 260

3.1 68xx/64xx Introduction

3.1.1 Overview

This device is highly integrated single-chip RADAR device in TI's 45-nm low-power RFCMOS technology.

3.1.1.1 Features

- Frequency-Modulated Continuous Wave Radio Frequency Transceiver With 60-64-GHz Band
- Supports Three Transmitter Chains and Four Receiver Chains
- Chirp Profiles With Programmable Period and Slope
- 40-MHz, 50-MHz, 80-MHz, and 100-MHz XTAL/OSC Reference Input Clock
- 12, 14, and 16-bit Real/Complex ADC With Variable Baseband ADC Sampling Rates up to 12.5 MHz at 12-bits Complex
- Cortex R4F at 200-MHz Application Processor for Control Functionality and Safety-Critical Algorithms
- 68xx only: C674x DSP at up to 600 MHz for RADAR Data Processing
- Cortex R4F– Radio Processor at 200 MHz for Continuous Monitoring and Calibration of Analog/RF Functionality
- On-Chip Multicore Debug Support
- Customer-Programmable Efuse Support
- Up to 768KB of L3 Shared Memory Support
- High-Performance Data Transfer With Multiple DMA and EDMA-TPCC Engines
- CAN-FD Support
- QSPI Serial Flash Support
- MIBSPI, SPI, I2C, and UART Serial Interfaces Support
- Hardware in Loop (HIL) Support
- Two-Lane Serial LVDS Interface Support
- AES, SHA, PKA, and TRNG Engines for Security
- 3 EPWM (Three Enhanced Pulse Width Modulator)
- Hardware Accelerator for FFT, Filtering, and CFAR Processing

3.1.2 Description

3.1.2.1 Block Diagram

Figure 3-1 shows the block diagram of the device.

Figure 3-1. 68xx Block Diagram

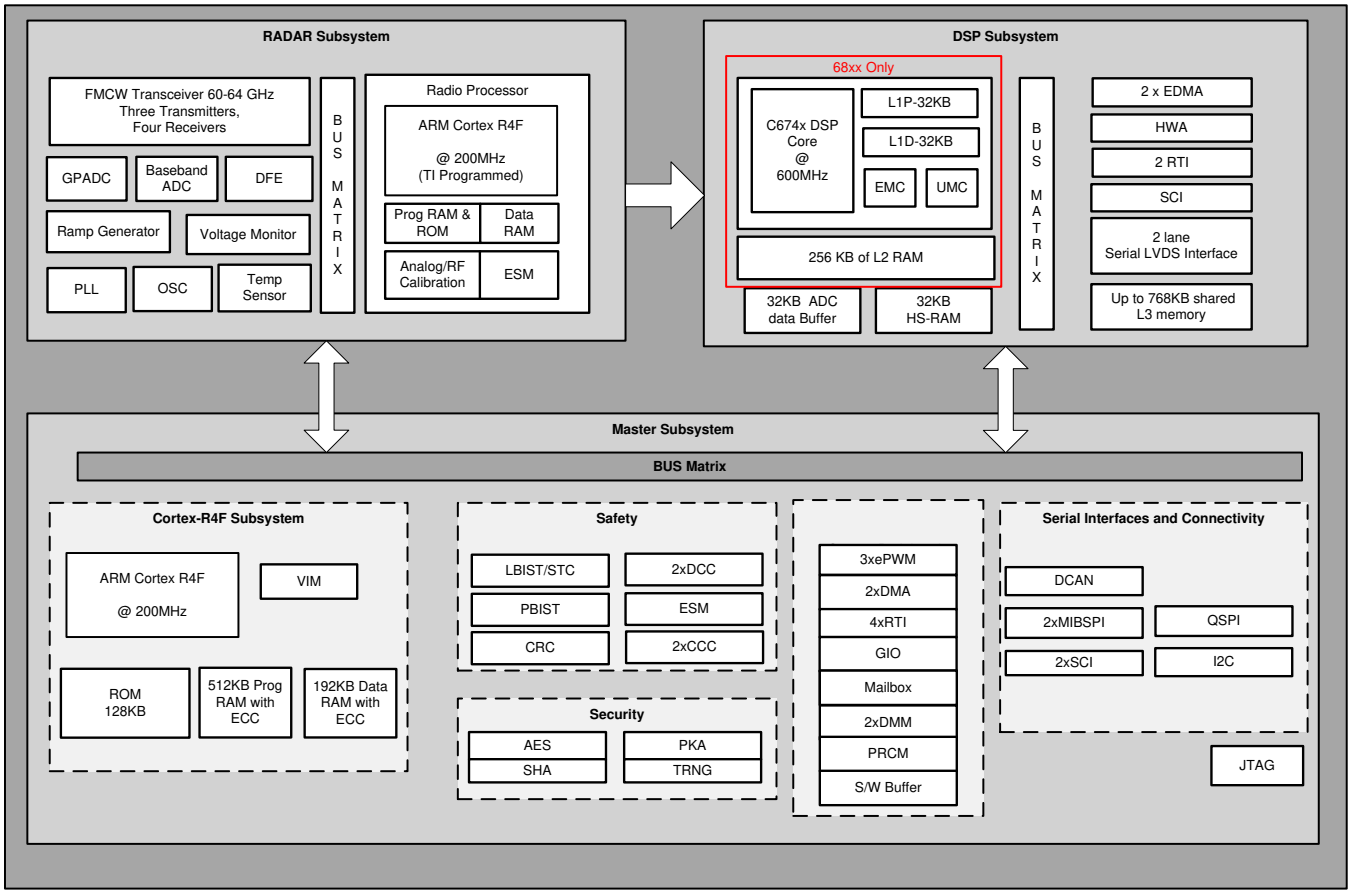


Table 3-1. Acronyms

ADC	Analog-to-Digital Convertor
AES	Advanced Encryption Standard
CRC	Cyclic Redundancy Check
DCAN	Controller Area Network
DCC	Digital Clock Comparator
DFE	Digital Front End
DMA	Direct Memory Access
DMM	Data Modification Module
ECC	Error Correcting Code
EDMA	Enhanced Direct Memory Access
EMC	Extended Memory Controller
ePWM	Enhanced Pulse Width Modulator
ESM	Error Signaling Module
GIO	General Input/Output
GPADC	General Purpose Analog-to-Digital Convertor
HS-RAM	Handshake RAM
HWA	Hardware Accelerator
I2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
L1D	Level 1 Data Memory Controller

Table 3-1. Acronyms (continued)

L1P	Level 1 Program Memory Controller
L2	Level 2
L3	Level 3
LBIST	Logic Built-In Self-Test
LVDS	Low Voltage Differential Signaling
MIBSPI	Multi-Buffered Serial Peripheral Interface Module
OSC	Oscillator
PBIST	Programmable Built-In Self-Test
PKA	Public Key Algorithm
PLL	Phase Locked Loops
PRCM	Power, Reset, Clock Management
QSPI	Quad Serial Peripheral Interface
RTI	Real Time Interrupt
S/W Buffer	Software Buffer
SCI	Serial Communication Interface
SHA	Secure Hash Algorithm
STC	Self-Test Controller
TRNG	True Random Number Generator
UMC	Unified Memory Controller
VIM	Vectored Interrupt Manager

3.1.2.2 Radar Subsystem

The RADAR subsystem is responsible for the RF and analog functionality of the device. The subsystem incorporates a built-in self-test processor for the continuous motoring and calibration of the analog and RF modules. The subsystem consists of:

- FMCW transceiver
  - Integrated PLL, transmitter, receiver, baseband, and A2D
  - 60-64-GHz coverage with 4-GHz available bandwidth
  - Four receive channels
  - Three transmit channels
  - Ultra-accurate chirp engine based on fractional-n PLL
  - 12,14, or 16-bit complex analog to digital converter
- Radio processor for built-in calibration and self-test
  - ARM Cortex R4F-based radio control system
  - Built-in firmware (ROM)
  - Self-calibrating system across frequency and temperature

This subsystem is TI-programmed with an API interface to the on-chip Cortex-R4F application processor.

3.1.2.3 DSP Subsystem (Applicable Devices Only)

The DSP subsystem consists of the following:

- The TMS320C674x™ VLIW DSP core from the generation, and the TMS320C64x+™ DSP architecture for RADAR data processing. These are enhancements from TI's C64x+™ DSP architecture, with additional features.
- 32KB L1D and 32KB of L1P cache/RAM
- 256KB of L2 RAM
- On-chip L3 shared memory of 768KB, with 256KB dedicated to DSP and 512 KB of memory shared

between the DSP and master subsystems.

- 32KB of memory for storing ADC samples from the RADAR subsystem
- Multiple Enhanced Direct Memory Access (EDMA) engines – TPCCs for high-performance data transfers
- 2-lane LVDS interface with support of up to 900 Mbps per lane for the RADAR raw ADC data transfer
- One watchdog timer and a general purpose timer implemented by the real time interrupt (RTI) modules
- One serial communication Interface (SCI) module implementing standard universal asynchronous receiver-transmitter (UART).
- Emulation capabilities
- Little Endian

3.1.2.4 Master Subsystem

The master subsystem consists of the following features:

- Cortex-R4F core supporting ARMv7-R, VFPv3-D16, and ARMv7 debug architecture
- Tightly-coupled memories
  - 128 KB of ROM
  - 512 KB of program RAM with ECC
  - 192KB of data RAM with ECC
- Hardware auto-initialization of the memories
- Vectored interrupt manager for prioritizing and controlling the interrupts for different sources

3.1.2.4.1 Serial Interfaces

- One MCAN controller supporting bit rates of up to 10 Mbit/s. Compliant to the controller area network (CAN) 2.0 part A, B protocol specification and ISO 11898-1, and CAN FD® V1.0 specification with up to 64 data bytes support.
- One I2C controller module with rates up to 400 kbps
- Two MIBSPI modules
- Two serial communication interface (SCI) modules implementing standard universal asynchronous receiver-transmitter (UART) with baud rates of up to 3.125 Mbps
- One quad SPI module support with maximum rate of 40 MHz

3.1.2.4.2 System Peripherals

- Multiple general-purpose input/output (GPIO) modules
- Direct memory access modules for high-performance data transfers
- One watchdog timer and a general purpose timer implemented by the real-time interrupt (RTI) modules
- Mailbox module for interprocessor communication
- Two data modification modules (DMM) with up to 65 Mbit/s data rate per pin
- Three enhanced pulse width (ePWM) modulator modules
- System reset and control module, which contains registers for the following functions:
  - Status
  - Efuse logic
  - I/O configuration
  - PAD configuration
  - System boot decoding logic

3.1.2.5 Functional Safety Deliverables

See the Device Safety Manual for supported features.

3.1.2.6 On-Chip Debug Support

The on-chip debug support has the following features:

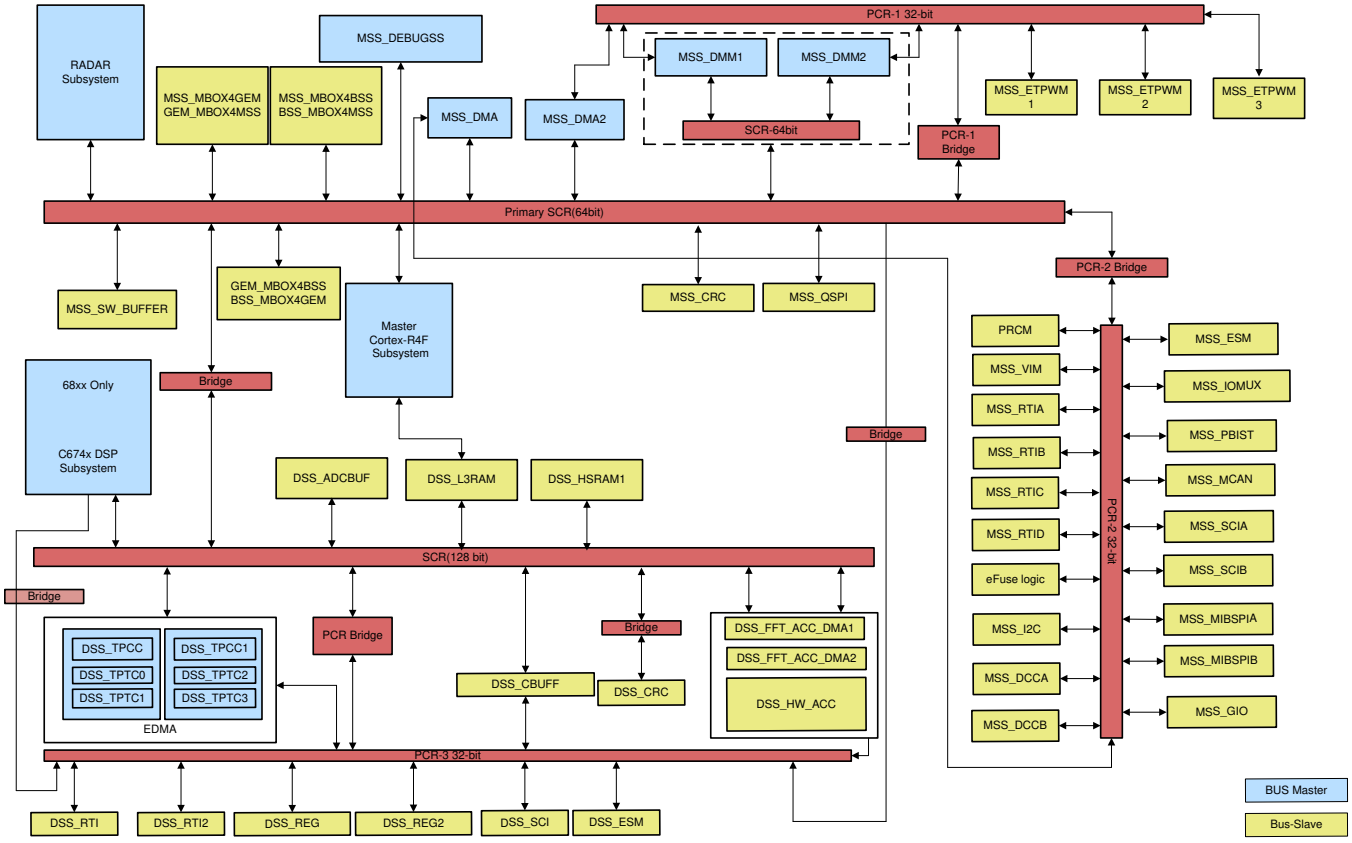
- Multiprocessor debugging to let users control multiple cores embedded in the device, such as:
  - Global starting and stopping of individual or multiple processors
  - Each processor can generate triggers to alter the execution flow of other processors
  - Interconnection of multiple devices
  - Channel triggering
- The following device cores can be debugged through Code Composer Studio (CCS):
  - Cortex-R4F
  - DSP
- Target debugging using IEEE1149.1 (JTAG®) port
- The debug subsystem includes:
  - IEEE1149.7 adapter
  - Generic TAP for emulation and test control (ICEPick-D™)
  - Debug access port (DAP)
  - Embedded trace macro (ETM)
  - Trace port interface Unit (TPIU)
  - Embedded trace buffer (ETB)

3.2 Memory Map

3.2.1 System Interconnect

The device implements a system interconnect based on TI's common bus architecture, comprising of VBUSM and VBUSP protocols. Figure 3-2 shows the interconnect diagram.

Figure 3-2. System Interconnect



The system interconnect is designed for the high-performance needs of the system. Its divided into three interconnect systems local to each of the three subsystems: the RADAR subsystem, DSP subsystem, and master subsystem. The interconnection of all these subsystems is shown in Figure 3-2.

In the master subsystem, the primary VBUSM SCR is responsible for managing the arbitration priority between accesses from multiple masters to each of the slaves. The arbitration priority is always round-robin.

The master subsystem has PCR interconnect that manages the accesses to the peripheral registers and peripheral memories, and provides a global reset for all peripherals. It also supports the capability to selectively enable or disable the clock for each peripheral individually. The PCR also manages the accesses to the system module registers required to configure the device clocks, interrupts, and so forth. The system module registers include status flags for indicating exception conditions – resets, aborts, errors, and interrupts.

Similarly, the 128-bit VBUSM SCR in the DSP subsystem manages the arbitration between accesses from the multiple masters to the slaves. The DSP subsystem has a 32-bit VBUSP PCR for the system and non-system peripherals.

3.2.2 Master Subsystem Cortex-R4F Memory Map

Table 3-2 shows the master subsystem, Cortex-R4F memory map.

Table 3-2. Master Subsystem, Cortex-R4F Memory Map

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_TCMA_ROM	0x0000_0000	0x0001_7FFF	128KiB	MSS_TCMA_ROM (TCMA) Program ROM (refer to ROM Eclipsing section)
Reserved	0x0001_8000	0x001F_FFFF		Reserved (refer to ROM Eclipsing section)
MSS_TCMA_RAM	0x0020_0000	0x07FF_FFFF	512KiB	MSS_TCMA_RAM (TCMA) size varies based on device and DSS_L3 (L3) sharing options configured (refer to ROM Eclipsing section)
MSS_TCMB	0x0800_0000	0x0C1F_FFFF	192KiB	MSS_TCMB (TCMB) Data RAM
MSS_SW_BUFFER	0x0C20_0000	0x0C20_1FFF	8KiB	MSS_SW_BUFFER (SWBUFF) Scratchpad memory
Reserved	0x0C20_2000	0x4FFF_FFFF		Reserved
DSS_TPTC0	0x5000_0000	0x5000_03FF	792B	DSS_TPTC0 (EDMA TPTC0) module configuration space
DSS_REG	0x5000_0400	0x5000_07FF	868B	DSS_REG (DSPSS) control module registers
DSS_TPTC1	0x5000_0800	0x5000_0BFF	792B	DSS_TPTC1 (EDMA TPTC1) module configuration space
DSS_REG2	0x5000_0C00	0x5000_FFFF	680B	DSS_REG2 (DSPSS) control module registers
DSS_TPCC	0x5001_0000	0x5001_FFFF	16KiB	DSS_TPCC (EDMA TPCC0) module configuration space
DSS_RTI	0x5002_0000	0x5002_FFFF	192B	DSS_RTI (WDT/RTI1) configuration space
DSS_SCI	0x5003_0000	0x5003_FFFF	148B	DSS_SCI (SCI) memory space
DSS_STC	0x5004_0000	0x5004_FFFF	284B	DSS_STC (STC) module configuration space (refer to Safety chapter)
Reserved	0x5005_0000	0x5006_FFFF		Reserved
DSS_CBUFF	0x5007_0000	0x5007_FFFF	564B	DSS_CBUFF (CBUFF) module configuration registers (refer to HSI chapter)
DSS_HW_ACC_PARAM	0x5008_0000	0x5008_01FF	512B	DSS_HW_ACC_PARAM (HWA) FFT accelerator PARAM memory
DSS_HW_ACC_MC_PING	0x5008_2000	0x5008_27FF	2 KiB	DSS_HW_ACC_MC_PING Registers
DSS_HW_ACC_MC_PONG	0x5008_3000	0x5008_37FF	2 KiB	DSS_HW_ACC_MC_PONG Registers
DSS_HW_ACC_STATIC	0x5008_0800	0x5008_0FFF	616B	DSS_HW_ACC_STATIC (HWA) FFT accelerator configuration registers

Table 3-2. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_HW_ACC_WIN	0x5008_1000	0x5008_FFFF	4KiB	DSS_HW_ACC_WIN (HWA) FFT accelerator Window registers
DSS_TPTC2	0x5009_0000	0x5009_03FF	792B	DSS_TPTC2 (EDMA TPTC2) module configuration space
DSS_TPTC3	0x5009_0400	0x5009_FFFF	792B	DSS_TPTC3 (EDMA TPTC3) module configuration space
DSS_TPCC1	0x500A_0000	0x500A_FFFF	16KiB	DSS_TPCC1 (EDMA TPCC1) module configuration space
Reserved	0x500B_0000	0x500C_FFFF		Reserved
DSS_ESM	0x500D_0000	0x500E_FFFF	1KiB	DSS_ESM (ESM) module configuration registers (refer to Safety chapter)
DSS_RTI2	0x500F_0000	0x500F_FFFF	192B	DSS_RTI2 (RTI2) module configuration registers
Reserved	0x5010_0000	0x50FF_FFFF		Reserved
DSS_L3RAM	0x5100_0000	0x51FF_FFFF	2MB	DSS_L3RAM (L3) shared memory space
DSS_ADCBUF	0x5200_0000	0x5201_FFFF	32KiB	DSS_ADCBUF (ADC) buffer memory space
DSS_CBUFF_FIFO	0x5202_0000	0x5202_7FFF	16KiB	DSS_CBUFF_FIFO (CBUFF) FIFO space (refer to HSI chapter)
Reserved	0x5202_8000	0x5202_FFFF		Reserved
DSS_FFT_ACC_DMA1	0x5203_0000	0x5203_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator Memory -1 space
DSS_FFT_ACC_DMA2	0x5203_8000	0x5206_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator Memory -2 space
Reserved	0x5207_0000	0x5207_FFFF		
DSS_HSRAM1	0x5208_0000	0x5208_FFFF	32KiB	DSS_HSRAM1 (HSRAM) Handshake memory space
Reserved	0x5209_0000	0x577D_FFFF		Reserved
DSS_DSP_L2_UMAP1	0x577E_0000	0x577F_FFFF	128KiB	DSS_DSP_L2_UMAP1 (L2) RAM space
DSS_DSP_L2_UMAP0	0x5780_0000	0x57DF_FFFF	128KiB	DSS_DSP_L2_UMAP0 (L2) RAM space
DSS_DSP_L1P	0x57E0_0000	0x57EF_FFFF	32KiB	DSS_DSP_L1P (L1) program memory space
DSS_DSP_L1D	0x57F0_0000	0xBFFF_FFFF	32KiB	DSS_DSP_L1D (L1) data memory space
EXT_FLASH	0xC000_0000	0xC07F_FFFF	8MB	MSS_QSPI (QSPI) flash memory space
MSS_QSPI	0xC080_0000	0xF060_0FFF	116B	MSS_QSPI (QSPI) module configuration registers
MSS_MBOX4BSS	0xF060_1000	0xF060_1FFF	2KiB	MSS_MBOX4BSS mailbox memory space
BSS_MBOX4MSS	0xF060_2000	0xF060_3FFF	2KiB	BSS_MBOX4MSS mailbox memory space

Table 3-2. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
GEM_MBOX4MSS	0xF060_4000	0xF060_4FFF	2KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0xF060_5000	0xF060_5FFF	2KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0xF060_6000	0xF060_6FFF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0xF060_7000	0xF060_7FFF	2KiB	BSS_MBOX4GEM mailbox memory space
BSS_MBOX4MSS_REG	0xF060_8000	0xF060_80FF	188B	BSS_MBOX4MSS_REG mailbox Configuration registers
BSS_MBOX4GEM_REG	0xF060_8100	0xF060_81FF	188B	BSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4BSS_REG	0xF060_8200	0xF060_82FF	188B	GEM_MBOX4BSS_REG mailbox Configuration registers
MSS_MBOX4GEM_REG	0xF060_8300	0xF060_83FF	188B	MSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4MSS_REG	0xF060_8400	0xF060_85FF	188B	GEM_MBOX4MSS_REG mailbox Configuration registers
MSS_MBOX4BSS_REG	0xF060_8600	0xFCF7_8BFF	188B	MSS_MBOX4BSS_REG mailbox Configuration registers
MSS_ETPWM1	0xFCF7_8C00	0xFCF7_8CFF	116B	MSS_ETPWM1 (ePWM1) configuration registers
MSS_ETPWM2	0xFCF7_8D00	0xFCF7_8DFF	116B	MSS_ETPWM2 (ePWM2) configuration registers
MSS_ETPWM3	0xFCF7_8E00	0xFCF8_0FFF	116B	MSS_ETPWM3 (ePWM3) configuration registers
MSS_DMA2_RAM	0xFCF8_1000	0xFCFF_0FFF	4KiB	MSS_DMA2_RAM (DMA2) RAM memory space
MSS_PCR2	0xFCFF_1000	0xFCFF_F5FF	1KiB	MSS_PCR2 (PCR_2) interconnect configuration port
MSS_RTID	0xFCFF_EC00	0xFCFF_ECBF	192B	MSS_RTID (RTID) module configuration registers
MSS_DMM2	0xFCFF_F600	0xFCFF_F6FF	472B	MSS_DMM2 (DMM2) module configuration registers
MSS_DMM	0xFCFF_F700	0xFCFF_F7FF	472B	MSS_DMM (DMM1) module configuration registers
MSS_DMA2_REG	0xFCFF_F800	0xFCFF_FFFF	1KiB	MSS_DMA2_REG (DMA2) module configuration registers
MSS_RTIC	0xFCFF FC00	0xFCFF FCBF	192B	MSS_RTIC (RTIC) module configuration registers
MSS_DTHE	0xFD00_0000	0xFD00_3FFF	3KiB	MSS_DTHE (Crypto) module configuration registers
Reserved	0xFD00_4000	0xFDFF_FFFF		Reserved
MSS_MCRC	0xFE00_0000	0xFF0B_FFFF	16KiB	MSS_MCRC (CRC) module configuration registers

Table 3-2. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_MIBSPIB_TXRAM	0xFF0C_0000	0xFF0C_01FF	0.5KiB	MSS_MIBSPIB_TXRAM (MIBSPIB) TX RAM memory space
MSS_MIBSPIB_RXRAM	0xFF0C_0200	0xFF0D_FFFF	0.5KiB	MSS_MIBSPIB_RXRAM (MIBSPIB) RX RAM memory space
MSS_MIBSPIA_TXRAM	0xFF0E_0000	0xFF0E_01FF	0.5KiB	MSS_MIBSPIA_TXRAM (MIBSPIA) TX RAM memory space
MSS_MIBSPIA_RXRAM	0xFF0E_0200	0xFF1D_FFFF	0.5KiB	MSS_MIBSPIA_RXRAM (MIBSPIA) RX RAM memory space
Reserved	0xFF1E_0000	0xFF4F_FFFF	68KiB	Reserved
MSS_MCAN_MSGMEM	0xFF50_0000	0xFF9F_FFFF	68KiB	MSS_MCAN_MSGMEM (MCAN) RAM memory space
MSS_DEBUGSS	0xFFA0_0000	0xFFF7_7FFF	244KiB	MSS_DEBUGSS (Debug subsystem) memory space and registers
MSS_PCR	0xFFF7_8000	0xFFF7_9FFF	1KiB	MSS_PCR (PCR_1) interconnect configuration port
MSS_MCAN_ECC	0xFFF7_A000	0xFFFF A20F	528B	MSS_MCAN_ECC (MCAN) module registers
Reserved	0xFFF7 A400	0xFFF7_BBFF	528B	Reserved
MSS_GIO	0xFFF7_BC00	0xFFF7_C7FF	180B	MSS_GIO (GIO) module configuration registers
MSS_MCAN_CFG	0xFFF7_C800	0xFFF7_D3FF	768B	MSS_MCAN_CFG (MCAN) module configuration registers
MSS_I2C	0xFFF7_D400	0xFFF7_DBFF	112B	MSS_I2C (I2C) module configuration registers
Reserved	0xFFF7_DC00	0xFFF7_E4FF		Reserved
MSS_SCIA	0xFFF7_E500	0xFFF7_E6FF	148B	MSS_SCIA (SCIA/UART) module configuration registers
MSS_SCIB	0xFFF7_E700	0xFFF7_F3FF	148B	MSS_SCIB (SCIB/UART) module configuration registers
MSS_MIBSPIA	0xFFF7_F400	0xFFF7_F5FF	512B	MSS_MIBSPIA (MIBSPIA) module configuration registers
MSS_MIBSPIB	0xFFF7_F600	0xFFF7_FFFF	512B	MSS_MIBSPIB (MIBSPIB) module configuration registers
MSS_DMA_RAM	0xFFF8_0000	0xFFF8_1FFF	4KiB	MSS_DMA_RAM (DMA1) RAM memory space
MSS_VIM_MEM	0xFFF8_2000	0xFFF8_BFFF	2KB	MSS_VIM_MEM (VIM) RAM memory space
Reserved	0xFFF8_C000	0xFFFF_E0FF		Reserved
MSS_TOPRCM	0xFFFF_E100	0xFFFF_E3FF	756B	MSS_TOPRCM TOP Level Reset, Clock management registers
MSS_PBIST	0xFFFF_E400	0xFFFF_E5FF	464B	MSS_PBIST (PBIST) module configuration registers (refer to Safety chapter)

Table 3-2. Master Subsystem, Cortex-R4F Memory Map (continued)

ModuleName	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_STC	0xFFFF_E600	0xFFFF_E9FF	284B	MSS_STC (STC) module configuration registers (refer to Safety chapter)
MSS_IOMUX	0xFFFF_EA00	0xFFFF_EBFF	512B	MSS_IOMUX (IOMUX) module registers
MSS_DCCA	0xFFFF_EC00	0xFFFF_EDFF	44B	MSS_DCCA (DCCA) module configuration registers
MSS_RTIB	0xFFFF_EE00	0xFFFF_EFFF	192B	MSS_RTIB (WDT/RTIB) module configuration registers
MSS_DMA_REG	0xFFFF_F000	0xFFFF_F3FF	1KiB	MSS_DMA_REG (DMA1) module configuration registers
MSS_DCCB	0xFFFF_F400	0xFFFF_F4FF	44B	MSS_DCCB (DCCB) module configuration registers
MSS_ESM	0xFFFF_F500	0xFFFF_F5FF	156B	MSS_ESM (ESM) module configuration registers (refer to Safety chapter)
Reserved	0xFFFF_F600	0xFFFF_F7FF		Reserved
MSS_GPCFG_REG	0xFFFF_F800	0xFFFF_FBFF	352B	MSS_GPCFG_REG (GPCFG) General purpose control registers
MSS_RTIA	0xFFFF_FC00	0xFFFF_FCFF	192B	MSS_RTIA (RTIA) module configuration registers
MSS_VIM	0xFFFF_FD00	0xFFFF_FEFF	512B	MSS_VIM (VIM) module configuration registers
MSS_RCM	0xFFFF_FF00	0xFFFF_FFFF	256B	MSS_RCM (RCM) Reset, Clock management registers

3.2.2.1 Radar Subsystem Interface

The RADAR subsystem is accessible through a set of TI-implemented high-level API calls by the application running on the master CR4F. For more information on the 68xx RADAR subsystem interface, see the mmWave SDK user guide.

3.2.3 DSP Subsystem Memory Map (Applicable Devices Only)

Table 3-3 shows the DSP C674x memory map.

Table 3-3. DSP C674x Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSP_L2_UMAP1	0x007E_0000	0x007F_FFFF	128KiB	DSP_L2_UMAP1 (L2) RAM space
DSP_L2_UMAP0	0x0080_0000	0x0081_FFFF	128KiB	DSP_L2_UMAP0 (L2) RAM space
DSP_L1P	0x00E0_0000	0x00E0_7FFF	32KiB	DSP_L1P (L1) program memory space
DSP_L1D	0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D (L1) data memory space
DSS_TPTC0	0x0200_0000	0x0200_03FF	1KiB	DSS_TPTC0 (EDMA TPTC0) module configuration space
DSS_REG	0x0200_0400	0x0200_07FF	868B	DSS_REG (DSPSS) control module registers
DSS_TPTC1	0x0200_0800	0x0200_0BFF	1KiB	DSS_TPTC1 (EDMA TPTC1) module configuration space

Table 3-3. DSP C674x Memory Map (continued)

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_REG2	0x0200_0C00	0x0200_0FFF	680B	DSS_REG2 (DSPSS) control module registers
DSS_TPCC	0x0201_0000	0x0201_3FFF	16KiB	DSS_TPCC (EDMA TPCC0) module configuration space
DSS_RTI	0x0202_0000	0x0202_00FF	192B	DSS_RTI (WDT/RTI1) module configuration registers
DSS_SCI	0x0203_0000	0x0203_00FF	148B	DSS_SCI (SCI/UART) module Configuration registers
DSS_CBUFF	0x0207_0000	0x0207_03FF	564B	DSS_CBUFF (CBUFF) module Configuration registers (refer to HSI chapter)
DSS_HW_ACC_PARAM	0x0208_0000	0x0208_07FF	512B	DSS_HW_ACC_PARAM (HWA) FFT accelerator PARAM memory
DSS_HW_ACC_STATIC	0x0208_0800	0x0208_0FFF	616B	DSS_HW_ACC_STATIC (HWA) FFT accelerator configuration registers
DSS_HW_ACC_WIN	0x0208_1000	0x0208_1FFF	4KiB	DSS_HW_ACC_WIN (HWA) FFT accelerator Window registers
DSS_HW_ACC_MC_PING	0x0208_2000	0x0208_27FF	2KiB	DSS_HW_ACC_MC_PING Registers
DSS_HW_ACC_MC_PONG	0x0208_3000	0x0208_37FF	2KiB	DSS_HW_ACC_MC_PONG Registers
DSS_TPTC2	0x0209_0000	0x0209_03FF	1KiB	DSS_TPTC2 (EDMA TPTC2) module configuration space
DSS_TPTC3	0x0209_0400	0x0209_07FF	1KiB	DSS_TPTC3 (EDMA TPTC3) module configuration space
DSS_TPCC1	0x020A_0000	0x020A_3FFF	16KiB	DSS_TPCC1 (EDMA TPCC1) module configuration space
Reserved	0x020B_0000	0x020C_FFFF		Reserved
DSS_ESM	0x020D_0000	0x020E_FFFF	92B	DSS_ESM (ESM) module Configuration registers (refer to Safety chapter)
DSS_RTI2	0x020F_0000	0x020F_00FF	192B	DSS_RTI2 (RTI2) module configuration registers
DSS_FFT_ACC_DMA1	0x2103_0000	0x2103_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator Memory -1 space
DSS_FFT_ACC_DMA2	0x2103_8000	0x2103_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator Memory -2 space
Reserved	0x0210_0000	0x0460_7FFF		Reserved
BSS_MBOX4MSS_REG	0x0460_8000	0x0460_80FF	188B	BSS_MBOX4MSS_REG mailbox Configuration registers
BSS_MBOX4GEM_REG	0x0460_8100	0x0460_81FF	188B	BSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4BSS_REG	0x0460_8200	0x0460_82FF	188B	GEM_MBOX4BSS_REG mailbox Configuration registers
MSS_MBOX4GEM_REG	0x0460_8300	0x0460_83FF	188B	MSS_MBOX4GEM_REG mailbox Configuration registers
GEM_MBOX4MSS_REG	0x0460_8400	0x0460_84FF	188B	GEM_MBOX4MSS_REG mailbox Configuration registers
MSS_MBOX4BSS_REG	0x0460_8600	0x0460_86FF	188B	MSS_MBOX4BSS_REG mailbox Configuration registers
Reserved	0x050C_0000	0x1FFF_FFFF		Reserved
DSS_L3RAM	0x2000_0000	0x201F_FFFF	2MB	DSS_L3RAM (L3) shared memory space
DSS_ADCBUF	0x2100_0000	0x2100_7FFC	32KiB	DSS_ADCBUF (ADC buffer) memory space
DSS_CBUFF_FIFO	0x2102_0000	0x2102_3FFC	16KiB	DSS_CBUFF_FIFO (Common buffer) FIFO space (Refer to HSI chapter)
Reserved	0x2102_8000	0x2107_FFFF		Reserved
DSS_HSRAM1	0x2108_0000	0x2108_7FFC	32KiB	DSS_HSRAM1 (HSRAM) Handshake memory space

Table 3-3. DSP C674x Memory Map (continued)

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
Reserved	0x2109_0000	0x21FF_FFFF		Reserved
DSS_MCRC	0x2200_0000	0x2200_03FF	1KiB	DSS_MCRC (CRC) module Configuration registers
Reserved	0x2500_0000	0x5060_0FFF		Reserved
MSS_MBOX4BSS	0x5060_1000	0x5060_17FF	2KiB	MSS_MBOX4BSS mailbox memory space
BSS_MBOX4MSS	0x5060_2000	0x5060_27FF	2KiB	BSS_MBOX4MSS mailbox memory space
GEM_MBOX4MSS	0x5060_4000	0x5060_47FF	2KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0x5060_5000	0x5060_57FF	2KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0x5060_6000	0x5060_67FF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_57FF	2KiB	BSS_MBOX4GEM mailbox memory space
Reserved	0x5600_0000	0x5060_5FFF		Reserved
GEM_MBOX4BSS	0x5060_6000	0x5060_67FF	2KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_7FFF	2KiB	BSS_MBOX4GEM mailbox memory space
Reserved	0x5600_0000	0xFFFF_FFFF		Reserved

3.2.4 EDMA Memory Map

Table 3-4 shows the EDMA-TPTC memory map.

Table 3-4. EDMA-TPTC Memory Map

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
DSS_EDMA_SCI	0x0603_0000	0x0603_00FF	148B	DSS_SCI memory space view from EDMA
DSS_DSP_L2_UMAP1	0x107E_0000	0x107F_FFFF	128KiB	DSP_L2_UMAP1 (L2) memory view from EDMA
DSS_DSP_L2_UMAP0	0x1080_0000	0x1081_FFFF	128KiB	DSP_L2_UMAP0 (L2) memory view from EDMA
DSS_DSP_L1P	0x10E0_0000	0x10E0_7FFF	32KiB	DSP_L1P (L1) program memory view from EDMA
DSS_DSP_L1D	0x10F0_0000	0x10F0_7FFF	32KiB	DSP_L1D (L1) data memory view from EDMA
DSS_L3RAM	0x2000_0000	0x201F_FFFF	2MB	DSS_L3RAM shared memory space
DSS_ADCBUF	0x2100_0000	0x2100_7FFC	32KiB	DSS_ADCBUF memory space
DSS_CBUFF_FIFO	0x2102_0000	0x2102_3FFC	16KiB	DSS_CBUFF_FIFO (Common buffer) memory space (Refer to HSI chapter)
DSS_FFT_ACC_DMA1	0x2103 0000	0x2103_7FFF	32KiB	DSS_FFT_ACC_DMA1 (HWA DMA) FFT accelerator memory-1 space
DSS_FFT_ACC_DMA2	0x2103 8000	0x2107_FFFF	32KiB	DSS_FFT_ACC_DMA2 (HWA DMA) FFT accelerator memory-2 space
DSS_HSRAM1	0x2108_0000	0x2108_7FFC	32KiB	DSS_HSRAM1 (Handshake) memory
MSS_TCMA_RAM	0x4020_0000	0x4023_FFFF	256 KiB	MSS_TCMA_RAM (TCMA) Data RAM
MSS_TCMB	0x4800_0000	0x4802_FFFF	192 KiB	MSS_TCMB (TCMB) Data RAM
MSS_SW_BUFFER	0x4C20_0000	0x4C20_1FFF	8 KiB	MSS_SW_BUFFER S/W Scratchpad memory
GEM_MBOX4MSS	0x5060_4000	0x5060_4000	2 KiB	GEM_MBOX4MSS mailbox memory space
MSS_MBOX4GEM	0x5060_5000	0x5060_5000	2 KiB	MSS_MBOX4GEM mailbox memory space
GEM_MBOX4BSS	0x5060_6000	0x5060_6000	2 KiB	GEM_MBOX4BSS mailbox memory space
BSS_MBOX4GEM	0x5060_7000	0x5060_7000	2 KiB	BSS_MBOX4GEM mailbox memory space

3.3 68xx/64xx Integration

3.3.1 Cortex-R4F Subsystem

3.3.1.1 Tightly Coupled Memories

Table 3-5 lists the dedicated MSS_TCMA_RAM and MSS_TCMB sizes for the Cortex R4F processor in the master subsystem, and also mentions the total available L3 shared RAM in the device. A portion of this L3 shared memory (DSS_L3RAM) can be allotted as TCM, to further increase the MSS_TCMA_RAM and MSS_TCMB available for the Cortex R4F.

Table 3-5. TCM and Shared Memory Available for Cortex R4F in Master Subsystem

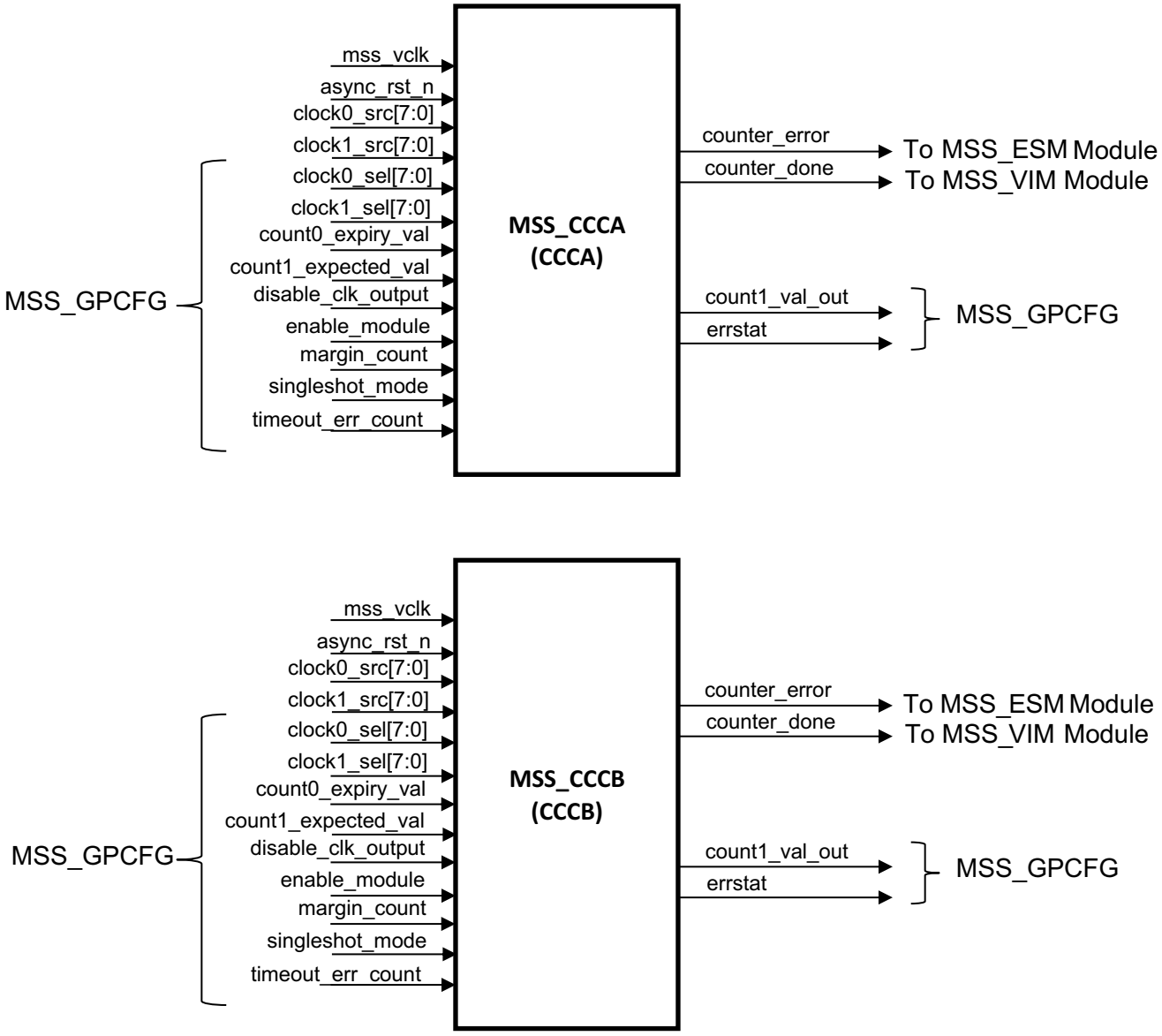
Cortex R4F		Shared
MSS_TCMA_RAM (Program RAM) (KB)	MSS_TCMB (Data RAM) (KB)	L3 Shared (KB)
512	192	768

See on how the L3 shared memory (DSS_L3RAM) can be assigned between the Cortex R4F of the master subsystem and the DSP core.

3.3.2 Clock Comparator

3.3.2.1 Core Clock Comparator (MSS_CCCA/MSS_CCCB)

Figure 3-3. Integration of MSS_CCCA and MSS_CCCB Modules



3.3.2.1.1 MSS_CCCA and MSS_CCCB Integration Connections

This device has two instances of CCC: MSS_CCCA (CCCA) and MSS_CCCB (CCCB). The clock connectivity information for these two instances is provided in Table 3-6. Configuration and status of this module is available through the MSS_GPCFG registers of the device.

Table 3-6. MSS_CCCA and MSS_CCCB Integration Connections

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
counter_error	ESM_GRP1[1]	ESM_GRP1[4]
counter_done	IRQ[80]	IRQ[81]

Table 3-6. MSS_CCCA and MSS_CCCB Integration Connections (continued)

	MSS_CCCA (CCCA)	MSS_CCCB (CCCB)
clock0_src[0]	REFCLK	CR4_VCLK
clock0_src[1]	CPUCLK	DSSCLK
clock0_src[2]	RCCLK	BSSCLK
clock0_src[3]	RCCLK	QSPICLK
clock0_src[4]	RCCLK	N/A
clock0_src[5]	RCCLK	CPUCLK
clock0_src[6]	RCCLK	REFCLK
clock0_src[7]	RCCLK	RCCLK
clock1_src[0]	REFCLK	PLLCLK_600
clock1_src[1]	PLLCLK_600	MSS_VCLK
clock1_src[2]	PLL 240Mhz	CPUCLK
clock1_src[3]	RCCLK	CR4_VCLK
clock1_src[4]	RCCLK	MSS_VCLK
clock1_src[5]	RCCLK	DSSCLK
clock1_src[6]	RCCLK	BSSCLK
clock1_src[7]	RCCLK	QSPICLK

3.3.2.2 Dual Clock Comparator (MSS_DCCA/MSS_DCCB)

Figure 3-4. MSS_DCCA/MSS_DCCB Integration Diagram

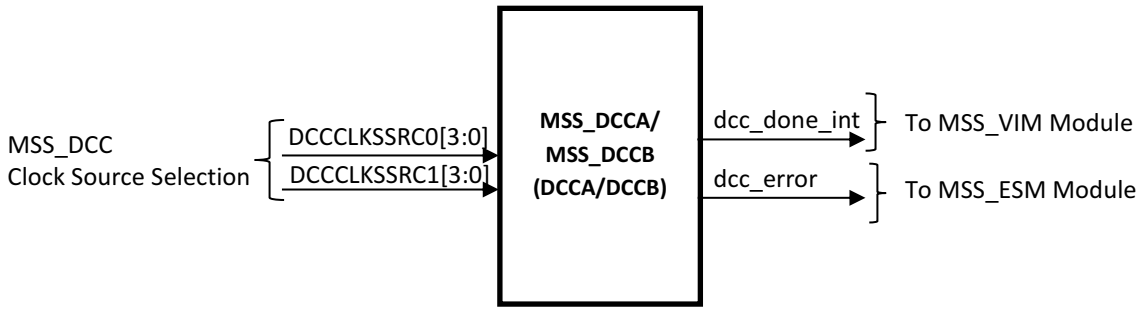


Table 3-7. MSS_DCCA Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - REF_CLK	0x0 - REF_CLK
0xA - PLL_600	0x1 - CPU_CLK
0x5 - PLL_240	0x2 to 0x7 - RC_CLK

Table 3-8. MSS_DCCB Clock Source Selection Table

DCCCLKSSRC0[3:0]	DCCCLKSSRC1[3:0]
0x0 - PLL_600	0x0 - VCLK
0xA - VCLK	0x1 - DSS_CLK
0x5 - CPU_CLK	0x2 - BSS_CLK
	0x3 - QSPI_CLK
	0x4 - Reserved
	0x5 - REF_CLK
	0x6 - CPU_CLK
	0x7 - RC_CLK

NOTE: Any values not mentioned are not used.

3.3.3 C674x DSP Subsystem (Applicable Devices Only)

3.3.3.1 DSP Event Assignment

Table 3-9. DSP Event Assignment

Event No.	Interrupt	Description
0	EVT0	Output of event combiner0, for events 1 through 31
1	EVT1	Output of event combiner0, for events 32 through 63
2	EVT2	Output of event combiner0, for events 64 through 95
3	EVT3	Output of event combiner0, for events 96 through 127
4	Reserved	Reserved
5	Reserved	Reserved
6	Reserved	Reserved
7	Reserved	Reserved
8	Reserved	Reserved
9	Reserved	Reserved
10	Reserved	Reserved
11	Reserved	Reserved
12	Reserved	Reserved
13	IDMAINT0	From DSP EMC, IDMA Channel 0 Interrupt
14	IDMAINT1	From DSP EMC, IDMA Channel 1Interrupt
15	Reserved	Reserved
16	DSS_TPTC0_IRQ_DONE	DSS_TPTC0 (EDMA TPTC0) completion interrupt
17	DSS_TPTC0_IRQ_ERR	DSS_TPTC0 (EDMA TPTC0) Error Interrupt
18	DSS_TPTC1_IRQ_DONE	DSS_TPTC1 (EDMA TPTC1) completion interrupt
19	DSS_TPTC1_IRQ_ERR	DSS_TPTC1 (EDMA TPTC1) Error Interrupt
20	DSS_TPCC_IRQ_DONE	DSS_TPCC (EDMA TPCC0) Global completion Interrupt
21	DSS_TPCC_IRQ_ERR	DSS_TPCC (EDMA TPCC0) Error Interrupt
22	DSS_CBUFF_IRQ	DSS_CBUFF (COMMON BUFFER) Interrupt
23	Reserved	Reserved
24	DSS_CBUFF_ERR_INTR	DSS_CBUFF (COMMON BUFFER) Error Interrupt
25	Reserved	Reserved
26	DSS_FRAME_START_IRQ/DSS_DMMSWIN T0/DSS_DMMSWINT39	Mux of VIN Frame start or BSS DFE Frame start.
27	DSS_CHIRP_AVAIL_IRQ/DSS_DMMSWINT2 /DSS_DMMSWINT43	Mux of VIN Chirp Available or DFE chirp available..
28	Reserved	Reserved
29	FFT_ACC_PARAM_DONE_INTR	DSS_HW_ACC FFT accelerator - param done interrupt
30	FFT_ACC_DONE_INTR	DSS_HW_ACC FFT accelerator - done interrupt
31	FFT_ACC_ACCESS_ERR	DSS_HW_ACC FFT accelerator - access error interrupt
32	DSS_ESM_LOW_PRIORITY	MSS_ESM_IRQ (Aggregate of MSS_ESM_GP1 errors)
33	DSS_MCRC_INT	MSS_MCRC (CRC) Interrupt
34	DSS_PROG_FILT_ERR	Error interrupt from Programmable filter indicating wrong programming of filter length exceeding the allowed range.
35	GEM_WAKEUP_SOURCE_FROM_DFT	Wakeup source from DFT module.
36	DSS_STC_DONE	Done indication from DSS_STC
37	DSP_PBIST_DONE	DSP_PBIST done indication from GEM

Table 3-9. DSP Event Assignment (continued)

Event No.	Interrupt	Description
38	Reserved	Reserved
39	Reserved	Reserved
40	Reserved	Reserved
41	Reserved	Reserved
42	Reserved	Reserved
43	Reserved	Reserved
44	Reserved	Reserved
45	Reserved	Reserved
46	DSS_DMMSWINT8	Interrupt from DSS_DMM configurable
47	DSS_DMMSWINT4	Interrupt from DSS_DMM configurable
48	Reserved	Reserved
49	Reserved	Reserved
50	Reserved	Reserved
51	Reserved	Reserved
52	Reserved	Reserved
53	Reserved	Reserved
54	Reserved	Reserved
55	Reserved	Reserved
56	Reserved	Reserved
57	Reserved	Reserved
58	DSS_MSS_SW0	DSS_MSS_SW interrupt
59	DSS_MSS_SW1	DSS_MSS_SW interrupt
60	DSS_DMMSWINT5	Interrupt from DSS_DMM configurable
61	DSS_DMMSWINT6	Interrupt from DSS_DMM configurable
62	DSS_BSS_SW1	Radar SS SW Interrupt 0
63	DSS_BSS_SW2	Radar SS SW Interrupt 1
64	DSS_TPTC2_IRQ_DONE	DSS_TPTC2 (EDMA TPTC2) completion interrupt
65	DSS_TPTC2_IRQ_ERR	DSS_TPTC2 (EDMA TPTC2) Error Interrupt
66	DSS_TPTC3_IRQ_DONE	DSS_TPTC3 (EDMA TPTC3) completion interrupt
67	DSS_TPTC3_IRQ_ERR	DSS_TPTC3 (EDMA TPTC3) Error Interrupt
68	DSS_TPCC1_IRQ_DONE	DSS_TPCC1 (EDMA TPCC1) Global completion Interrupt
69	DSS_TPCC1_IRQ_ERR	DSS_TPCC1 (EDMA TPCC1) Error Interrupt
70	DSS_ADC_DATA_VALID_FALL/DSS_DMMS WINT3/DSS_DMMSWINT44	DSS_ADC Ping/Pong interrupt
71	DSS_UART_REQ0	DSS_SCI (UART) Req 0
72	DSS_UART_REQ1	DSS_SCI (UART) Req 1
73	DSS_RTI0_OVERFLOW_0	DSS_RTI Overflow 0
74	DSS_RTI0_OVERFLOW_1	DSS_RTI Overflow 1
75	DSS_RTI0_0	DSS_RTI Interrupt 0
76	DSS_RTI0_1	DSS_RTI Interrupt 1
77	DSS_RTI0_2	DSS_RTI Interrupt 2
78	DSS_RTI0_3	DSS_RTI Interrupt 3
79	DSS_RTI1_OVERFLOW_0	DSS_RTI2 Overflow 0
80	DSS_RTI1_OVERFLOW_1	DSS_RTI2 Overflow 1
81	DSS_RTI1_0	DSS_RTI2 Interrupt 0
82	DSS_RTI1_1	DSS_RTI2 Interrupt 1
83	DSS_RTI1_2	DSS_RTI2 Interrupt 2
84	DSS_RTI1_3	DSS_RTI2 Interrupt 3

Table 3-9. DSP Event Assignment (continued)

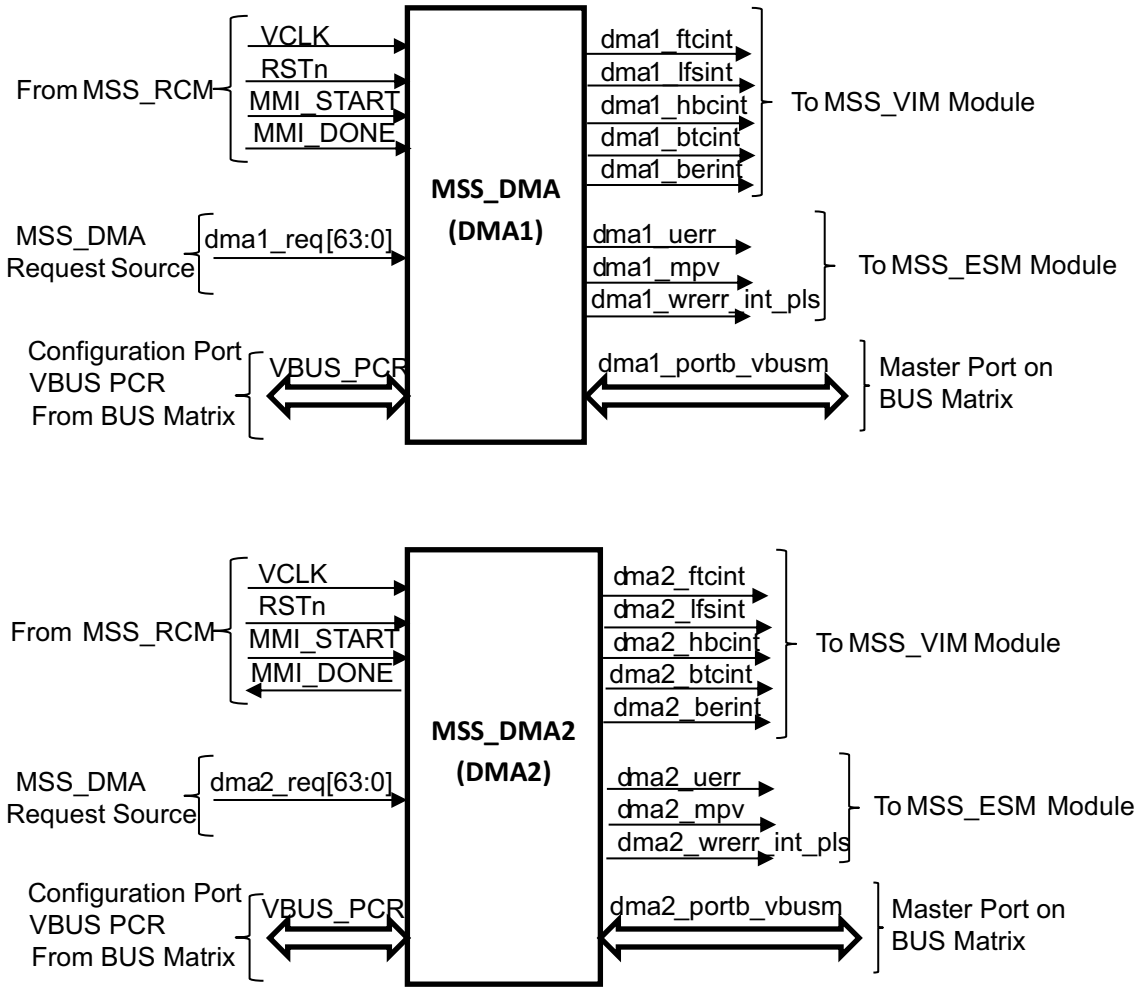
Event No.	Interrupt	Description
85	DSS_BSS_MAILBOX_FULL	Interrupt indicating there is a message from MSS in the Mailbox BSS-DSS
86	DSS_BSS_MAILBOX_EMPTY	Interrupt indicating the MSS has read/ack the message DSP posted in the Mailbox DSS-BSS
87	GPIO_0_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
88	GPIO_1_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
89	GPIO_2_host_interrupt	MSS_GIO (GPIO) host Interrupt Controller
90	Reserved	Reserved
91	DSS_MSS_MAILBOX_FULL	Interrupt indicating there is a message from MSS in the Mailbox MSS-DSS
92	DSS_MSS_MAILBOX_EMPTY	Interrupt indicating the MSS has read/ack the message DSP posted in the Mailbox DSS-MSS
93	DSS_LOGICAL_FRAME_START/DSS_DMM SWINT1/DSS_DMMSWINT40	Logical Frame start interrupt
94	DSS_DMMSWINT7	Interrupt from DSS_DMM configurable
95	Reserved	Reserved
96	INTERR	DSP dropped CPU interrupt event
97	IDMA_ERR	Invalid IDMA parameters
98	Reserved	Reserved
99	Reserved	Reserved
100	Reserved	Reserved
101	Reserved	Reserved
102	Reserved	Reserved
103	Reserved	Reserved
104	Reserved	Reserved
105	Reserved	Reserved
106	Reserved	Reserved
107	Reserved	Reserved
108	Reserved	Reserved
109	Reserved	Reserved
110	Reserved	Reserved
111	Reserved	Reserved
112	Reserved	Reserved
113	DSP_PMC_ED	DSS_DSP_L1P parity error
114	Reserved	Reserved
115	Reserved	Reserved
116	DSP_UMC_ED1	DSS_DSP_L2 ECC single error correction
117	DSP_UMC_ED2	DSS_DSP_L2 ECC double error detection
118	DSP_PDC_INT	Power down sleep interrupt
119	DSP_SYS_CMPA	CPU memory protection fault
120	DSP_L1P_CMPA	DSS_DSP_L1P CPU memory protection fault
121	DSP_L1P_DMPA	DSS_DSP_L1P DMA memory protection fault
122	DSP_L1D_CMPA	DSS_DSP_L1D CPU memory protection fault
123	DSP_L1D_DMPA	DSS_DSP_L1D DMA memory protection fault
124	DSP_L2_CMPA	DSS_DSP_L2 CPU memory protection fault
125	DSP_L2_DMPA	DSS_DSP_L2 DMA memory protection fault
126	DSP_EMC_CMPA	From EMC, CPU memory protection fault
127	DSP_EMC_BUSSERR	From EMC, bus error interrupt

3.3.4 Direct Memory Access Controller (MSS_DMA)

3.3.4.1 MSS_DMA Integration Diagrams

The device has two instances of DMA module, MSS_DMA and MSS_DMA2. Integration of the two DMA blocks in the device are shown in Figure 3-5 and .

Figure 3-5. Integration of MSS_DMA and MSS_DMA2 Module



3.3.4.2 MSS_DMA Features

- 64-bit OCP protocol to perform bus master accesses
- INCR-4 64-bit burst accesses
- Multithreading architecture allowing data of two different channel transfers to be interleaved during non-burst accesses
- 2-port configuration for parallel bus master
- Channels can be assigned to either high-priority queue or low-priority queue. Within each queue, fixed or round-robin priorities can be serviced
- Built-in ECC generation and evaluation logic for internal RAM-storing channel transfer information
- Supports multiple interrupt outputs for mapping to multiple interrupt controllers in multicore systems
- 48 requests can be mapped to any 32 channels
- Supports LE endianness

- External ECC Gen/Eval block of MSS_DMA support ECC generation for data transactions, and parity for address, and control signals (following Cortex-R5F standard)
- 8 MPU regions
- Channel-chaining capability
- Hardware and software MSS_DMA requests
- 8-, 16-, 32-, or 64-bit transactions supported
- Multiple addressing modes for source and destination (fixed, increment, offset)
- Auto-initiation

3.3.4.3 MSS_DMA Request Map

Both instances of MSS_DMA have 64 lines of request and are connected to identical input triggers, as shown in Table 3-10. This allows the two DMAs to trigger different types of transfers for the same request.

Table 3-10. MSS_DMA Request Map

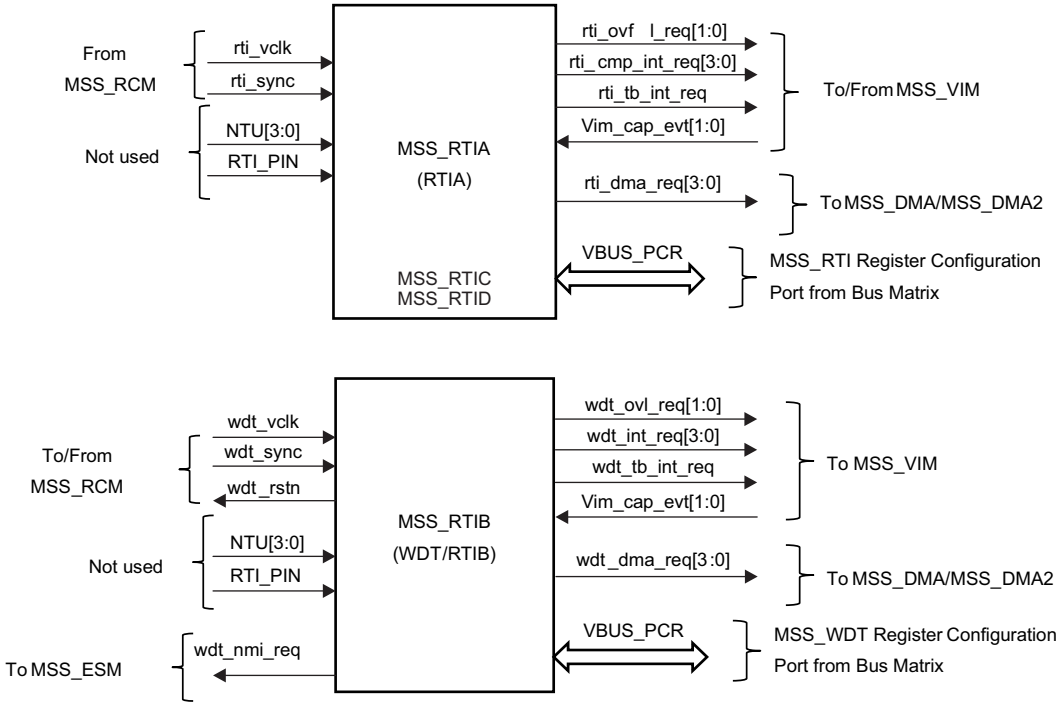
Module	DMA Request Sources	DMA Request	Mux Setting Register
MSS_MIBSPIA	MSS_MIBSPIA Channel-1	DMAREQ[0]	
MSS_MIBSPIA	MSS_MIBSPIA Channel-0	DMAREQ[1]	
MSS_MIBSPIB	MSS_MIBSPIB	DMAREQ[2]	
MSS_MIBSPIB	MSS_MIBSPIB	DMAREQ[3]	
MSS_QSPI	MSS_QSPI DMA request	DMAREQ[4]	
MSS_MIBSPIA	MSS_MIBSPIA Channel-3	DMAREQ[5]	
Reserved	Reserved	DMAREQ[6]	
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) DMAREQ	DMAREQ[7]	
Reserved	Reserved	DMAREQ[8]	
MSS_MIBSPIA	MSS_MIBSPIA Channel-5	DMAREQ[9]	
MSS_I2C	MSS_I2C receive	DMAREQ[10]	
MSS_I2C	MSS_I2C transmit	DMAREQ[11]	
MSS_RTIA	MSS_RTIA DMAREQ0	DMAREQ[12]	
MSS_RTIA	MSS_RTIA DMAREQ1	DMAREQ[13]	
Reserved	Reserved	DMAREQ[14]	
Reserved	Reserved	DMAREQ[15]	
Reserved	Reserved	DMAREQ[16]	
MSS_MIBSPIA	MSS_MIBSPIA Channel-2	DMAREQ[17]	
MSS_RTIA	MSS_RTIA DMAREQ2	DMAREQ[18]	
MSS_RTIA	MSS_RTIA DMAREQ3	DMAREQ[19]	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ0	DMAREQ[20]	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) DMAREQ1	DMAREQ[21]	
MSS_MIBSPIA	MSS_MIBSPIA Channel-4	DMAREQ[22]	
MSS_ETPWM3A	MSS_ETPWM3A DMAREQ	DMAREQ[23]	
MSS_RTIB (WDT/RTIB)	WDT/RTIB DMAREQ2	DMAREQ[24]	
MSS_RTIB (WDT/RTIB)	WDT/RTIB DMAREQ3	DMAREQ[25]	
MSS_MCRC (CRC)	MSS_MCRC (CRC) DMAREQ0	DMAREQ[26]	
MSS_MCRC (CRC)	MSS_MCRC (CRC) DMAREQ1	DMAREQ[27]	
MSS_SCIB (UART2)	MSS_SCIB (UART2) receive	DMAREQ[28]	
MSS_SCIB (UART2)	MSS_SCIB (UART2) transmit	DMAREQ[29]	
MSS_SCIA (UART1)	MSS_SCIA (UART1) receive	DMAREQ[30]	
MSS_SCIA (UART1)	MSS_SCIA (UART1) transmit	DMAREQ[31]	

Table 3-10. MSS_DMA Request Map (continued)

Module	DMA Request Sources	DMA Request	Mux Setting Register
MSS_GIO	MSS_GIO-0	DMAREQ[32]	
MSS_GIO	MSS_GIO-1	DMAREQ[33]	
MSS_GIO	MSS_GIO-2	DMAREQ[34]	
MSS_ETPWM1A	MSS_ETPWM1A DMAREQ	DMAREQ[35]	
Reserved	Reserved	DMAREQ[36]	
MSS_MIBSPIB	MSS_MIBSPIB Channel-2	DMAREQ[37]	GPCFG6[11] ,default - spib
MSS_RTIC	MSS_RTIC dma req[0]		
MSS_MIBSPIB	MSS_MIBSPIB Channel-3	DMAREQ[38]	GPCFG6[12] ,default - spib
MSS_RTIC	MSS_RTIC dma req[1]		
MSS_ETPWM1B	MSS_ETPWM1B DMAREQ	DMAREQ[39]	
MSS_ETPWM2A	MSS_ETPWM2A DMAREQ	DMAREQ[40]	
MSS_ETPWM2B	MSS_ETPWM2B DMAREQ	DMAREQ[41]	
MSS_MIBSPIB	MSS_MIBSPIB Channel-4	DMAREQ[42]	GPCFG6[13] ,default - spib
MSS_RTID	MSS_RTID dma req[0]		
MSS_MIBSPIB	MSS_MIBSPIB Channel-5	DMAREQ[43]	GPCFG6[14] ,default - spib
MSS_RTID	MSS_RTID dma req[1]		
Reserved	Reserved	DMAREQ[44]	
MSS_ETPWM3B	MSS_ETPWM3B DMAREQ	DMAREQ[45]	
MSS_GIO	MSS_GIO-14	DMAREQ[46]	
MSS_GIO	MSS_GIO-15	DMAREQ[47]	
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-0	DMAREQ[48]	
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-1	DMAREQ[49]	
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-2	DMAREQ[50]	
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-3	DMAREQ[51]	
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-4	DMAREQ[52]	
MSS_DTHE (Crypto/SHA)	SHA DMAREQ-5	DMAREQ[53]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-0	DMAREQ[54]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-1	DMAREQ[55]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-2	DMAREQ[56]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-3	DMAREQ[57]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-4	DMAREQ[58]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-5	DMAREQ[59]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-6	DMAREQ[60]	
MSS_DTHE (Crypto/AES)	AES DMAREQ-7	DMAREQ[61]	
MSS_MCAN (MCAN/CAN_FD)	MSS_MCAN (MCAN/CAN_FD) DMA channel-1	DMAREQ[62]	
MSS_MCAN (MCAN/CAN_FD)	MSS_MCAN (MCAN/CAN_FD) DMA channel-0	DMAREQ[63]	

3.3.5 Real Time Interrupt (MSS_RTIA) and RTI With Digital Watchdog Timer (MSS_RTIB)

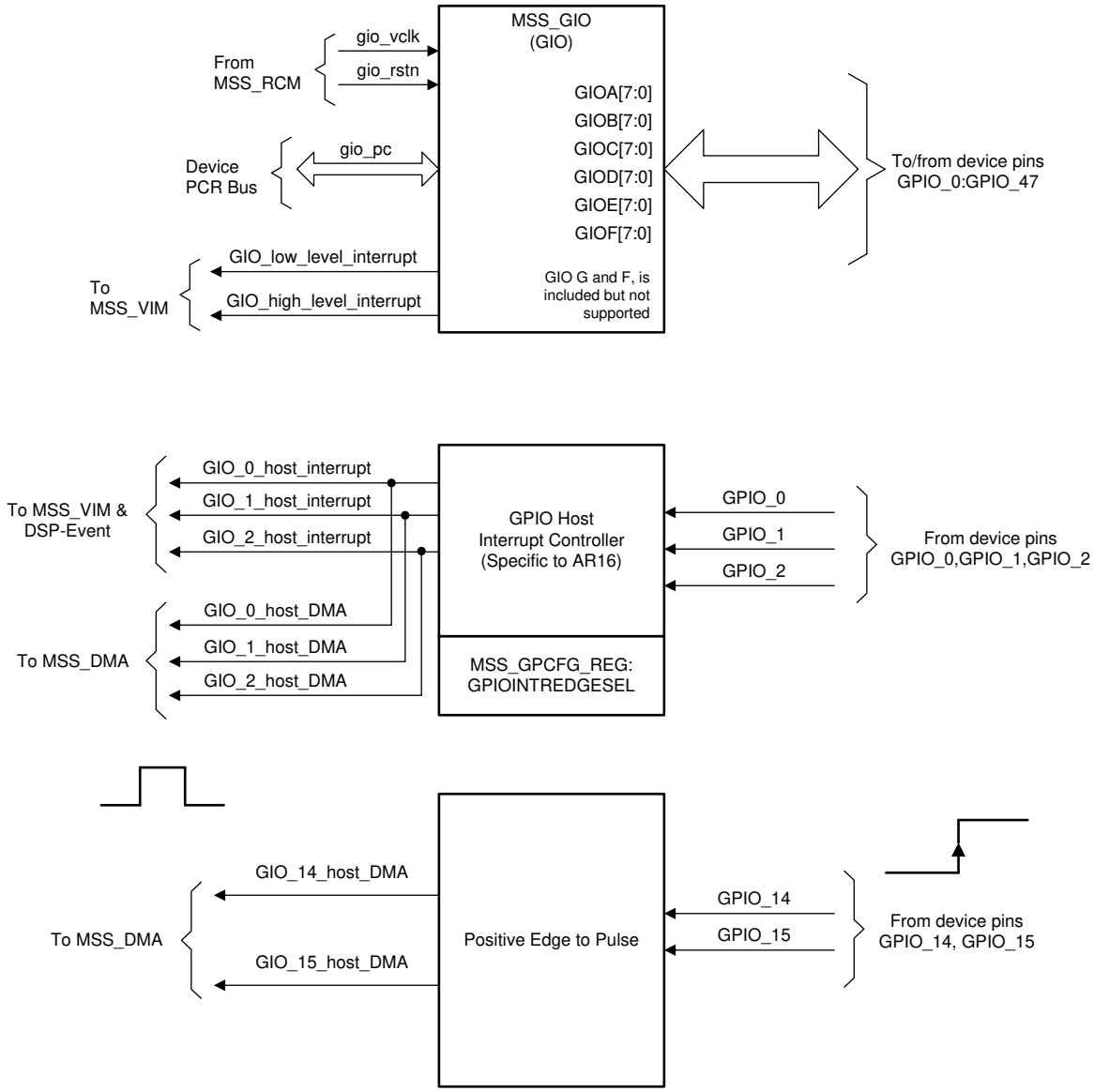
Figure 3-6. Integration of MSS_RTIA and MSS_RTIB, WDT Using the MSS_RTIB Module



3.3.6 General Purpose I/0 (MSS_GIO)

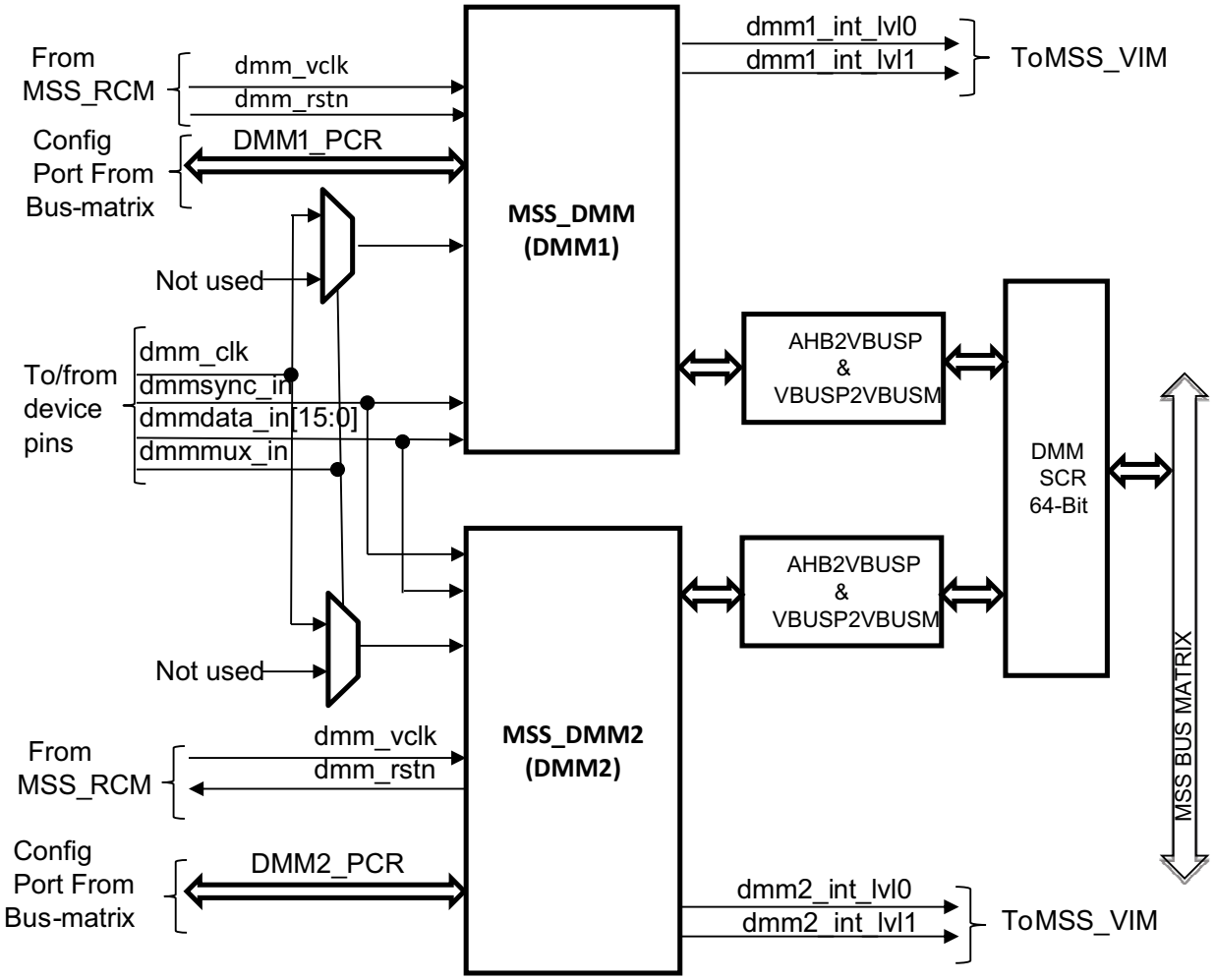
NOTE: Emulation mode and power-down mode (low-power mode) are not supported in the 68xx device.

Figure 3-7. Integration Block Diagram for MSS_GIO



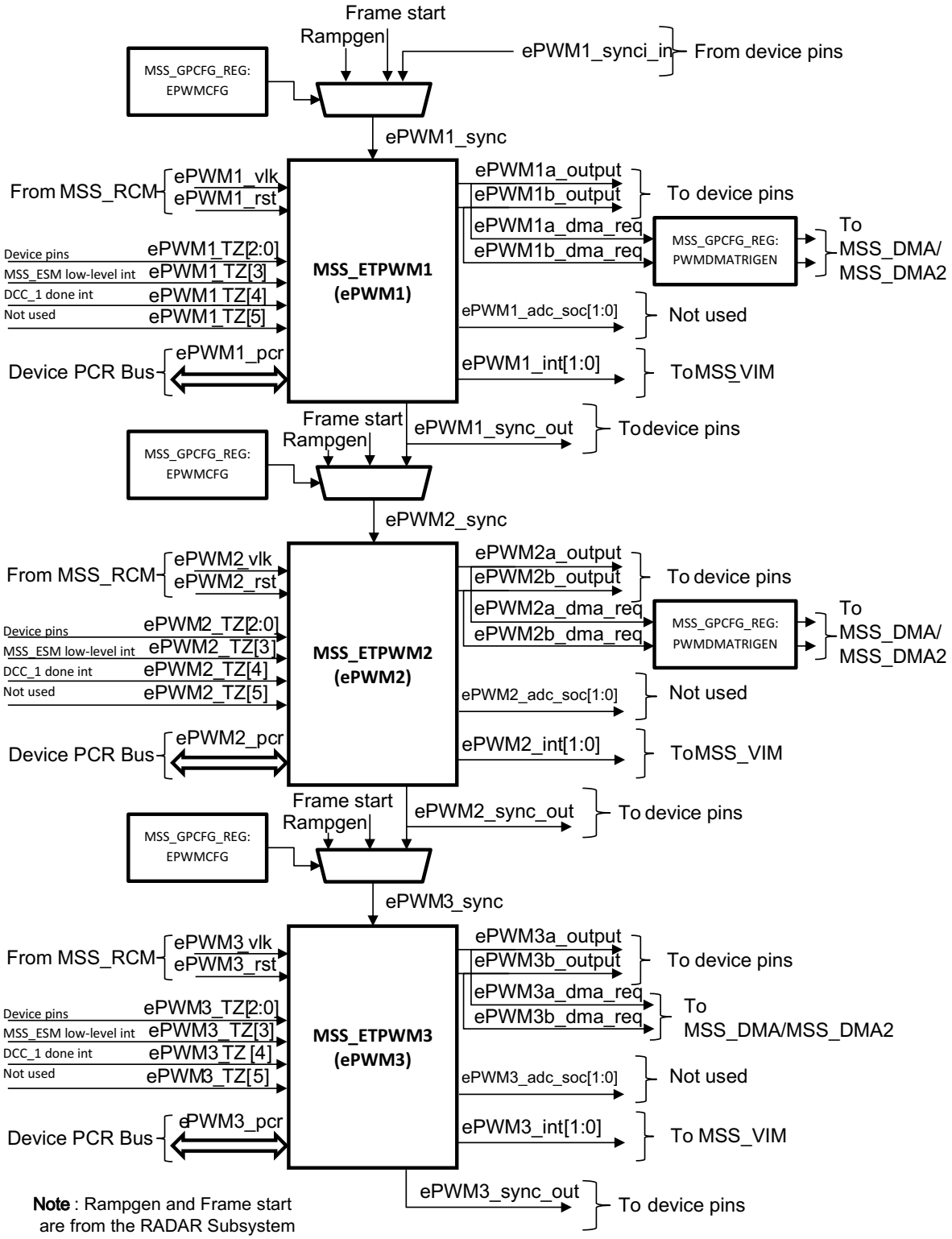
3.3.7 Data Modification Module (MSS_DMM)

Figure 3-8. MSS_DMM Integration



3.3.8 Enhanced Pulse Width Modulator (MSS_ETPWM)

Figure 3-9. Multiple MSS_ETPWM Modules



3.3.9 Vectored Interrupt Manager (MSS_VIM)

3.3.9.1 Interrupt Request Assignments

Table 3-11. Interrupt Request Assignments

Module	VIM Interrupt Sources	Default VIM Interrupt Channel	Mux Setting Register
MSS_ESM	MSS_ESM high-level interrupt(NMI)	0	
Reserved	Reserved	1	
MSS_RTIA	MSS_RTIA compare interrupt 0	2	
MSS_RTIA	MSS_RTIA compare interrupt 1	3	
MSS_RTIA	MSS_RTIA compare interrupt 2	4	
MSS_RTIA	MSS_RTIA compare interrupt 3	5	
MSS_RTIA	MSS_RTIA overflow interrupt 0	6	
MSS_RTIA	MSS_RTIA overflow interrupt 1	7	
MSS_RTIA	MSS_RTIA time-base	8	
MSS_GIO	MSS_GIO high-level interrupt	9	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 0	10	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt1	11	
MSS_MIBSPIA	MSS_MIBSPIA level 0 interrupt	12	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 2	13	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) interrupt 3	14	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) overflow interrupt 0	15	
Reserved	Reserved	16	
MSS_MIBSPIB	MSS_MIBSPIB level 0 Interrupt	17	
MSS_GIO host interrupt module	MSS_GIO GPIO_0_host_interrupt	18	
MSS_MCRC (CRC)	MSS_MCRC (CRC) interrupt	19	
MSS_ESM	MSS_ESM low-level interrupt	20	
SYSTEM	Software-triggered interrupt 4	21	
MSS Cortex R4F	MSS Cortex R4F interrupt PMU	22	
MSS_GIO	MSS_GIO low-level interrupt	23	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) overflow interrupt 1	24	
MSS_RTIB (WDT/RTIB)	MSS_RTIB (WDT/RTIB) TB base interrupt	25	
MSS_MIBSPIA	MSS_MIBSPIA level 0 interrupt	26	
MSS_QSPI	MSS_QSPI interrupt	27	
MSS_DMM	MSS_DMM S/W interrupt 38	28	
Reserved	Reserved	29	
MSS_MIBSPIB	MSS_MIBSPIB level 1 interrupt	30	
MSS_DTHE (Crypto/SHA)	MSS_DTHE (Crypto/SHA) SHA -S interrupt	31	
MSS_GIO host interrupt module	MSS_GIO GPIO_1_host_interrupt	32	
MSS_DMA	MSS_DMA frame transfer complete interrupt	33	
MSS_DMA	MSS_DMA last frame transfer start interrupt	34	
MSS_MCAN (MCAN/CAN_FD)	MSS_MCAN (MCAN/CAN_FD) interrupt-0	35	
MSS_DMM	MSS_DMM level -0 interrupt	36	
MSS_DTHE (Crypto/SHA)	MSS_DTHE (Crypto/SHA) SHA -P interrupt	37	
MSS_DTHE (Crypto/TRNG)	MSS_DTHE (Crypto/TRNG) TRNG interrupt	38	
MSS_DMA	MSS_DMA half-block transfer complete interrupt	39	

Table 3-11. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel	Mux Setting Register
MSS_DMA	MSS_DMA block transfer complete interrupt	40	
MSS_DMA2	MSS_DMA2 frame block transfer complete interrupt	41	
MSS_MCAN (MCAN/CAN_FD)		42	
MSS_DMM	MSS_DMM level -1 interrupt	43	
MSS_MCAN (MCAN/CAN_FD)	MSS_MCAN (MCAN/CAN_FD) message filter interrupt-0	44	
MSS_DMA2	MSS_DMA2 last frame complete interrupt	45	
MSS_MCAN (MCAN/CAN_FD)	MSS_MCAN (MCAN/CAN_FD) message filter interrupt-1	46	
FPU	Floating point unit interrupt	47	
MSS_GIO (GPIO host interrupt module)	MSS_GIO GPIO_2_host_interrupt	48	
MSS_DMA2	MSS_DMA2 half-block transfer complete interrupt	49	
MSS_DMA2	MSS_DMA2 block transfer complete interrupt	50	
MSS_DMA2	MSS_DMA2 bus error interrupt	51	
System	DSS to MSS software-triggered by register DSS_REG2:MSSSWIRQ:MSSSWIRQ1	52	
MSS_DTHE (Crypto/PKA)	MSS_DTHE (Crypto/PKA) PKA module interrupt	53	
MSS_DTHE (Crypto/AES)	MSS_DTHE (Crypto/AES) AES-S module interrupt	54	
MSS_MCAN (MCAN/CAN_FD)	MSS_MCAN (MCAN/CAN_FD) message filter interrupt-2	55	
MSS_DTHE (Crypto/AES)	MSS_DTHE (Crypto/AES) AES-P module interrupt	56	
MSS_DMM2	MSS_DMM2 level -0 interrupt	57	
MSS_DMM2	MSS_DMM2 level -1 interrupt	58	
Mailbox	DSS to MSS mailbox full interrupt	59	
Mailbox	DSS to MSS mailbox empty interrupt	60	
System	DSS to MSS software-triggered by register DSS_REG2:MSSSWIRQ:MSSSWIRQ2	61	
MSS_DEBUGSS (Debug subsystem)	MSS_DEBUGSS (Debug subsystem) interrupt	62	
DSPSS-MSS_STC	GEM MSS_STC done interrupt	63	
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 0 interrupt	64	
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 0 interrupt	65	
MSS_I2C	MSS_I2C interrupt	66	
MSS_DMM	MSS_DMM interrupt 34	67	
MSS_DMM	MSS_DMM interrupt 35	68	
MSS_DMM	MSS_DMM interrupt 36	69	
MSS_DMA	MSS_DMA bus error interrupt	70	
MSS_DMM/Radar subsystem	MSS_DMM interrupt 30 or Radar subsystem logical Frame Start	71	
Reserved	Reserved	72	
MSS_DMM	MSS_DMM interrupt 33	73	
MSS_SCIA (UART1)	MSS_SCIA (UART1) level 1 interrupt	74	
MSS_SCIB (UART2)	MSS_SCIB (UART2) level 1 interrupt	75	
SYSTEM	Software-triggered interrupt 0	76	
SYSTEM	Software-triggered interrupt 1	77	

Table 3-11. Interrupt Request Assignments (continued)

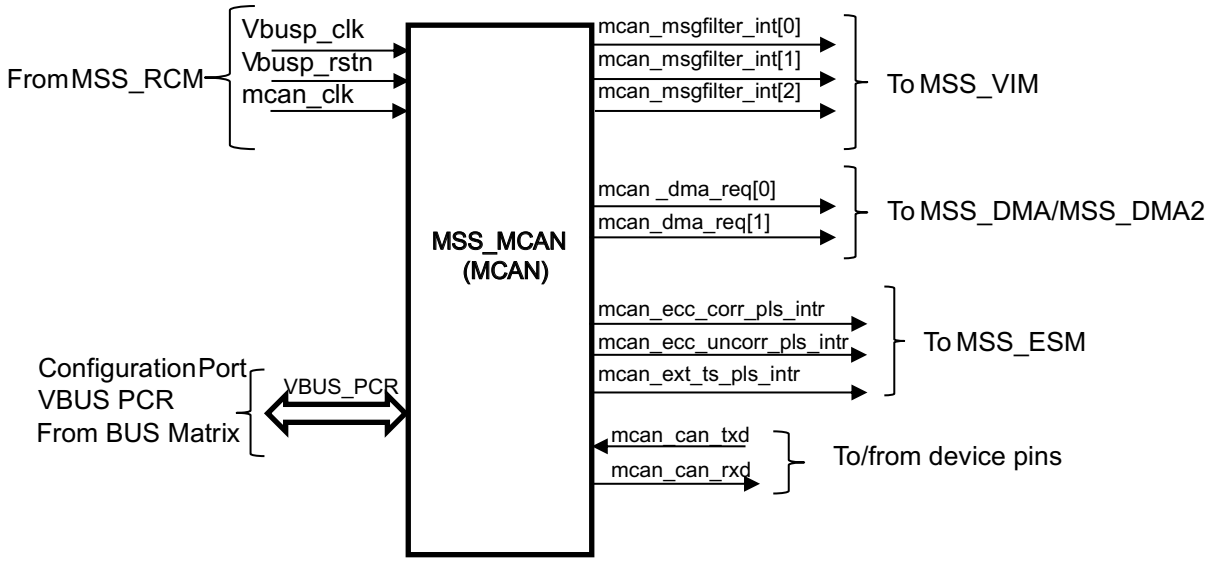
Module	VIM Interrupt Sources	Default VIM Interrupt Channel	Mux Setting Register
SYSTEM	Software-triggered interrupt 2	78	
SYSTEM	Software-triggered interrupt 3	79	
Reserved	Reserved	80	
Reserved	Reserved	81	
MSS_DCCA	MSS_DCCA (dual clock compare) module1-done interrupt	82	
MSS_DCCB	MSS_DCCB (dual clock compare) module2-done interrupt	83	
SYSTEM	Software-triggered interrupt 5	84	
MSS_PBIST	MSS_PBIST interrupt	85	
MSS_DMM/DSS	GEM IRQ-7/MSS_DMM interrupt 32	86	
Reserved	Reserved	87	
Reserved	Reserved	88	
Reserved	Reserved	89	
Reserved	Reserved	90	
Reserved	Reserved	91	
Reserved	Reserved	92	
Reserved	Reserved	93	
Reserved	Reserved	94	
MAILBOX	RADARSS to MSS mailbox interrupt	95	
MAILBOX	RADARSS mailbox read complete interrupt sent from RADARSS to MSS	96	
RADARSS	ADC valid fall interrupt	97	
MSS_DMM/RADARSS	MSS_DMM interrupt 29/frame start interrupt/	98	
RADARSS	Chirp start interrupt	99	
RADARSS	Chirp end Interrupt	100	
RADARSS	Frame end Interrupt	101	
Reserved	Reserved	102	
Reserved	Reserved	103	
MSS_ETPWM1	ePWM1 interrupt-1	104	
RADARSS-MSS_STC	MSS_STC done Interrupt	105	
RadarSS	All RadarSS interrupts combined	106	
MSS_ETPWM1	ePWM1 interrupt-2	107	
MSS_ETPWM2	ePWM2 interrupt-1	108	GPCFG6[26] , default - epwm
MSS_RTIC	MSS_RTIC interrupt 0		
MSS_ETPWM2	ePWM2 interrupt-2	109	GPCFG6[27] , default - epwm
MSS_RTIC	MSS_RTIC interrupt 1		
MSS_ETPWM3	ePWM3 interrupt-1	110	GPCFG6[28] , default - epwm
MSS_RTID	MSS_RTID interrupt 0		
MSS_ETPWM3	ePWM3 interrupt-2	111	GPCFG6[29] , default - epwm
MSS_RTID	MSS_RTID interrupt 1		
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) interrupt	112	
DSS_TPTC0 (EDMA TPTC0)	DSS_TPTC0 (EDMA TPTC0) error interrupt	113	
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) interrupt	114	

Table 3-11. Interrupt Request Assignments (continued)

Module	VIM Interrupt Sources	Default VIM Interrupt Channel	Mux Setting Register
DSS_TPTC1 (EDMA TPTC1)	DSS_TPTC1 (EDMA TPTC1) error interrupt	115	
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) interrupt	116	
DSS_TPCC (EDMA TPCC0)	DSS_TPCC (EDMA TPCC0) error interrupt	117	
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) interrupt	118	
Reserved	Reserved	119	
DSS_CBUFF (Common Buffer)	DSS_CBUFF (Common Buffer) error interrupt	120	
MSS_DMM	MSS_DMM interrupt 37	121	
Reserved	Reserved	122	
DSS_ADCBUF/MSS_DMM	Chirp available interrupt/MSS_DMM interrupt 31	123	
MSS_PBIST	MSS_PBIST: Gem MSS_STC done	124	
DSS_HW_ACC	DSS_HW_ACC FFT accelerator -param done interrupt	125	
DSS_HW_ACC	DSS_HW_ACC FFT accelerator - done interrupt	126	
DSS_HW_ACC	DSS_HW_ACC FFT accelerator –access error interrupt	127	

3.3.10 Module Controller Area Network (MSS_MCAN)

Figure 3-10. Integration Block Diagram for Module Controller Area Network (MSS_MCAN)



3.3.11 Multi-Buffered Serial Peripheral Interface Module (MSS_MIBSPI)

Figure 3-11. MSS_MIBSPIA Integration

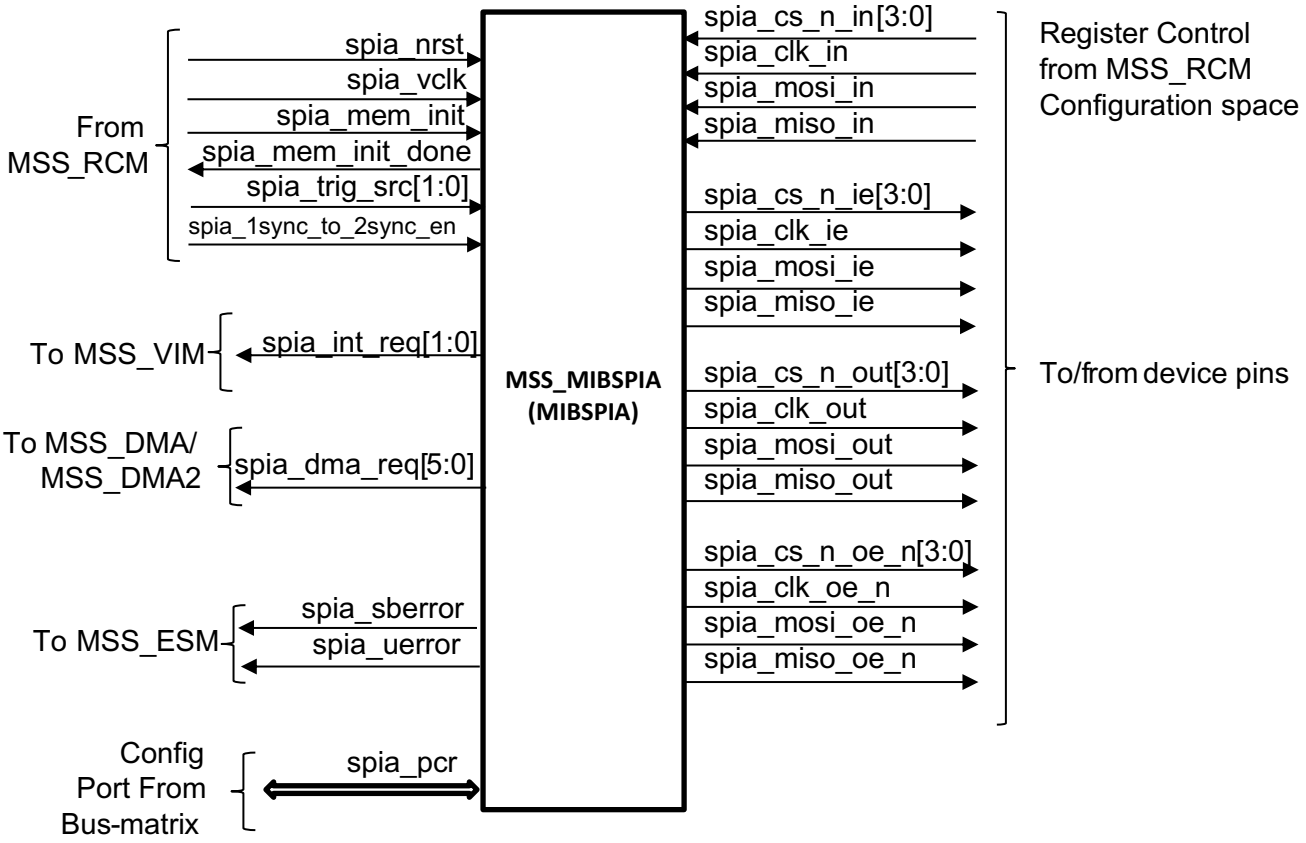
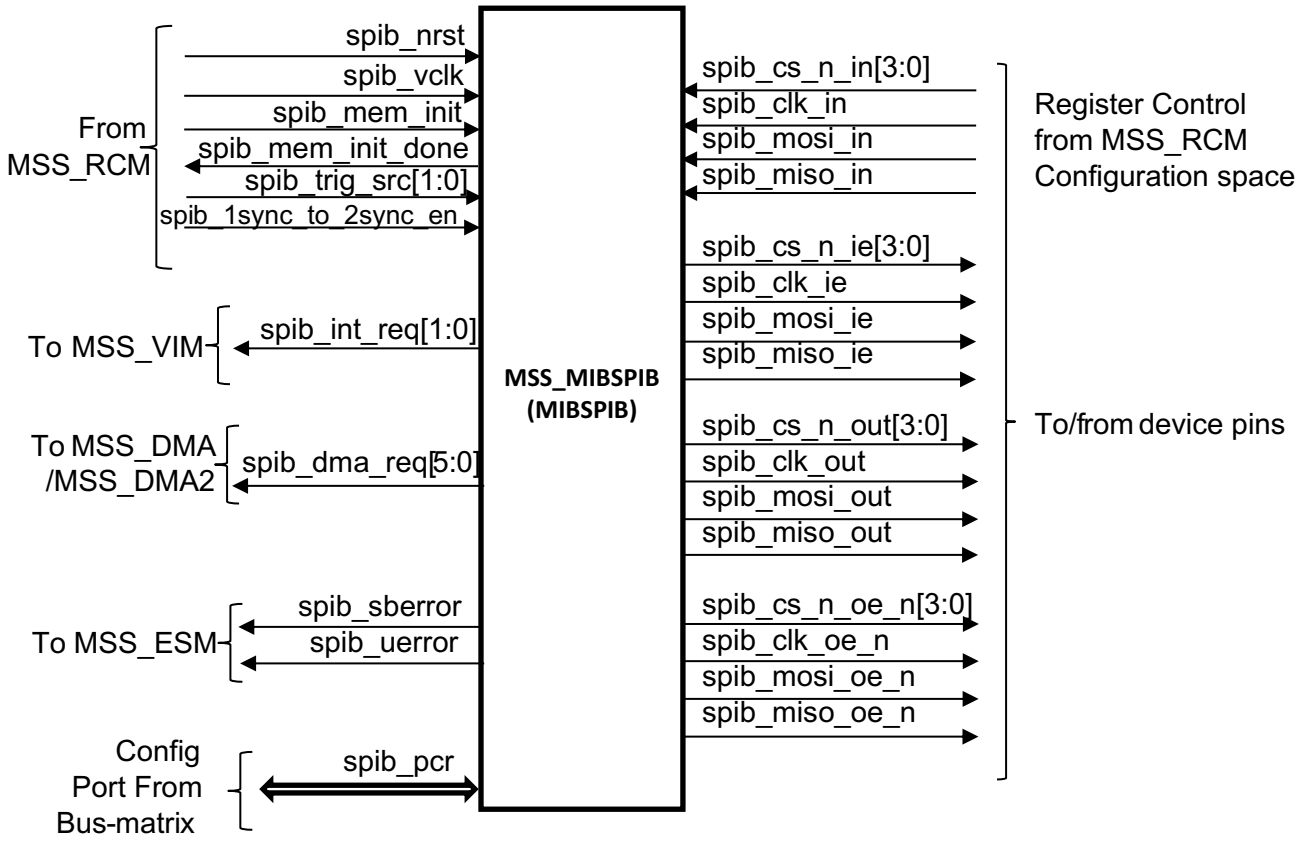


Figure 3-12. MSS_MIBSPIB Integration



3.3.12 Quad Serial Peripheral Interface (MSS_QSPI)

The MSS_QSPI module of the the device only supports one CS pin, (qspi1_cs).

3.3.13 Enhanced Direct Memory Access (EDMA)

3.3.13.1 EDMA Controller Integration

This device has two EDMA channel controllers (DSS_TPCC0 and DSS_TPCC1) on the device:

- DSS_TPCC0 (EDMA TPCC0) has two transfer controllers: DSS_TPTC0 (EDMA TPTC0) and DSS_TPTC1 (EDMA TPTC1)
- DSS_TPCC1 (EDMA TPCC1) has two transfer controllers: DSS_TPTC2 (EDMA TPTC2) and DSS_TPTC3 (EDMA TPTC3)

NOTE: This device does not support the region interrupt feature of the EDMA peripheral. Only the global interrupt feature of the EDMA module is supported.

Table 3-12. DSS_TPCC Configuration

	DSS_TPCC0 (EDMA TPCC0)	DSS_TPCC1 (EDMA TPCC1)
Number of MSS_DMA channels	64	64
Number of PaRAM entires	128	256
Number of QDMA channels	8	8

Table 3-12. DSS_TPCC Configuration (continued)

	DSS_TPCC0 (EDMA TPCC0)	DSS_TPCC1 (EDMA TPCC1)
Number of event queues	2	2
Memory protection existence	No	No
Channel mapping	No	No
Number of TCs (transfer controllers)	2	2

Table 3-13. DSS_TPTC Configuration

	DSS_TPTC[0-1]	DSS_TPTC[2-3]
FIFO size	512 bytes	128 bytes
TR pipe depth	2	2
Bus width	16 bytes	16 bytes

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 3-13 and Figure 3-14 show the EDMA controller integration.

Figure 3-13. EDMA Controller Integration (1 of 2)

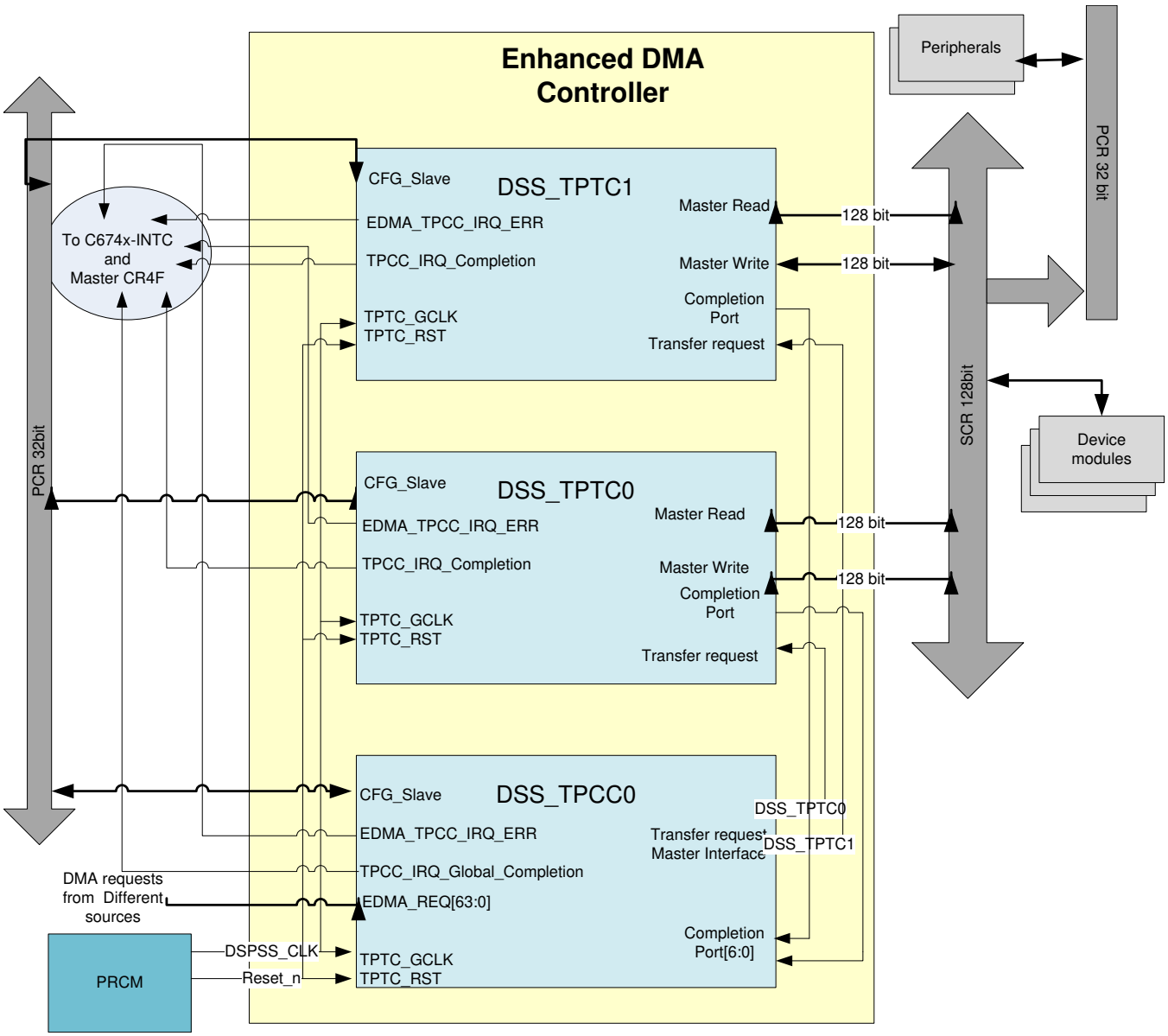
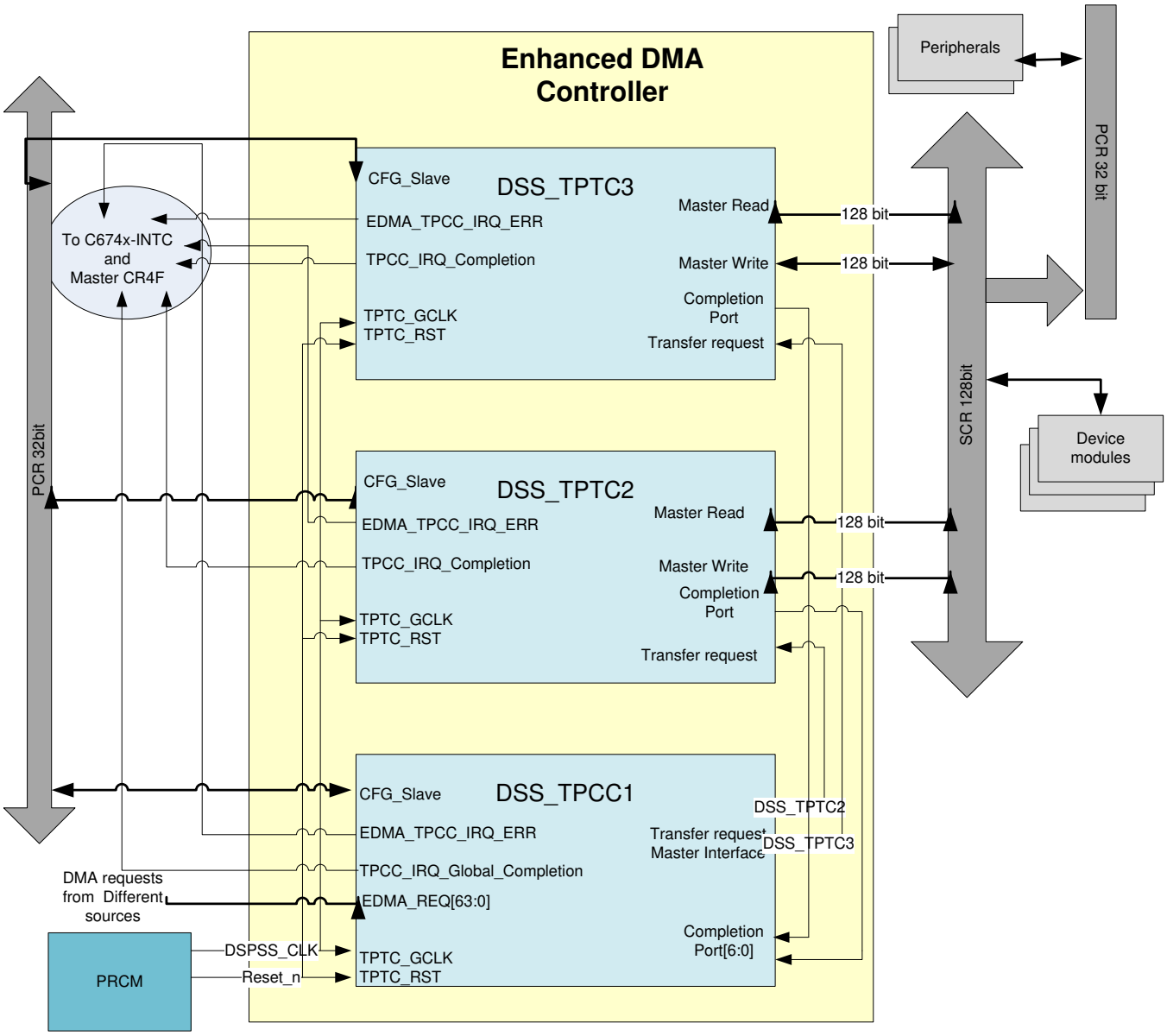


Figure 3-14. EDMA Controller Integration (2 of 2)



3.3.13.2 EDMA Request Map

Table 3-14. EDMA Request Map

Request Number	Hardware Event
DSS_TPCC0 (EDMA TPCC0) DMA	
0	DSS_CBUFF_DMA_REQ_0
1	DSS_CBUFF_DMA_REQ_1
2	DSS_CBUFF_DMA_REQ_2
3	DSS_CBUFF_DMA_REQ_3
4	DSS_CBUFF_DMA_REQ_4
5	DSS_CBUFF_DMA_REQ_5
6	DSS_CBUFF_DMA_REQ_6
7	RESERVED

Table 3-14. EDMA Request Map (continued)

Request Number	Hardware Event
8	Frame Start/DSS_DMMSWINT9/DSS_DMMSWINT39
9	Chirp Available/DSS_DMMSWINT11/DSS_DMMSWINT43
10	RESERVED
11	RESERVED
12	RESERVED
13	RESERVED
14	RESERVED
15	RESERVED
16	RESERVED
17	DSS_FFT_ACC_CHANNEL_TRIGGER_0
18	DSS_FFT_ACC_CHANNEL_TRIGGER_1
19	DSS_FFT_ACC_CHANNEL_TRIGGER_2
20	DSS_FFT_ACC_CHANNEL_TRIGGER_3
21	DSS_FFT_ACC_CHANNEL_TRIGGER_4
22	DSS_FFT_ACC_CHANNEL_TRIGGER_5
23	DSS_FFT_ACC_CHANNEL_TRIGGER_6
24	DSS_FFT_ACC_CHANNEL_TRIGGER_7
25	DSS_FFT_ACC_CHANNEL_TRIGGER_8
26	DSS_FFT_ACC_CHANNEL_TRIGGER_9
27	DSS_FFT_ACC_CHANNEL_TRIGGER_10
28	DSS_FFT_ACC_CHANNEL_TRIGGER_11
29	DSS_FFT_ACC_CHANNEL_TRIGGER_12
30	DSS_FFT_ACC_CHANNEL_TRIGGER_13
31	DSS_FFT_ACC_CHANNEL_TRIGGER_14
32	DSS_FFT_ACC_CHANNEL_TRIGGER_15
33	DSS_MCRC_DMA_REQ_0
34	DSS_MCRC_DMA_REQ_1
35	FRC_EVENT_GEN_0
36	FRC_EVENT_GEN_1
37	FRC_EVENT_GEN_2
38	FRC_EVENT_GEN_3
39	RESERVED
40	LOGICAL_FRAME_START/DSS_DMMSWINT10/DSS_DMMSWINT40
41	ADC_DATA_VALID_FALL/DSS_DMMSWINT12/DSS_DMMSWINT44
42	UART_DMA_REQ_0
43	UART_DMA_REQ_1
44	DMMSW_INT_13
45	DMMSW_INT_14
46	DMMSW_INT_15
47	DMMSW_INT_16
48	DMMSW_INT_17
49	GPIO_0_host_interrupt
50	GPIO_1_host_interrupt
51	GPIO_2_host_interrupt
52	RTI1_DMA_REQ_0
53	RTI1_DMA_REQ_1
54	RTI1_DMA_REQ_2

Table 3-14. EDMA Request Map (continued)

Request Number	Hardware Event
55	RTI1_DMA_REQ_3
56	RTI2_DMA_REQ_0
57	RTI2_DMA_REQ_1
58	RTI2_DMA_REQ_2
59	RTI2_DMA_REQ_3
60	RESERVED
61	RESERVED
62	RESERVED
63	DMMSW_INT_18
DSS_TPCC1 (EDMA TPCC1) MSS_DMA	
0	DSS_CBUFF_DMA_REQ_0
1	DSS_CBUFF_DMA_REQ_1
2	DSS_CBUFF_DMA_REQ_2
3	DSS_CBUFF_DMA_REQ_3
4	DSS_CBUFF_DMA_REQ_4
5	DSS_CBUFF_DMA_REQ_5
6	DSS_CBUFF_DMA_REQ_6
7	RESERVED
8	FRAME_START/DSS_DMMSWINT19/DSS_DMMSWINT39
9	CHIRP_AVIALABLE/DSS_DMMSWINT21/DSS_DMMSWINT43
10	RESERVED
11	RESERVED
12	RESERVED
13	RESERVED
14	RESERVED
15	RESERVED
16	RESERVED
17	DSS_FFT_ACC_CHANNEL_TRIGGER_0
18	DSS_FFT_ACC_CHANNEL_TRIGGER_1
19	DSS_FFT_ACC_CHANNEL_TRIGGER_2
20	DSS_FFT_ACC_CHANNEL_TRIGGER_3
21	DSS_FFT_ACC_CHANNEL_TRIGGER_4
22	DSS_FFT_ACC_CHANNEL_TRIGGER_5
23	DSS_FFT_ACC_CHANNEL_TRIGGER_6
24	DSS_FFT_ACC_CHANNEL_TRIGGER_7
25	DSS_FFT_ACC_CHANNEL_TRIGGER_8
26	DSS_FFT_ACC_CHANNEL_TRIGGER_9
27	DSS_FFT_ACC_CHANNEL_TRIGGER_10
28	DSS_FFT_ACC_CHANNEL_TRIGGER_11
29	DSS_FFT_ACC_CHANNEL_TRIGGER_12
30	DSS_FFT_ACC_CHANNEL_TRIGGER_13
31	DSS_FFT_ACC_CHANNEL_TRIGGER_14
32	DSS_FFT_ACC_CHANNEL_TRIGGER_15
18	RESERVED
19	RESERVED
20	RESERVED
21	RESERVED

Table 3-14. EDMA Request Map (continued)

Request Number	Hardware Event
22	RESERVED
23	RESERVED
24	RESERVED
25	RESERVED
26	RESERVED
27	RESERVED
28	RESERVED
29	RESERVED
30	RESERVED
31	RESERVED
32	RESERVED
33	DSS_MCRC_DMA_REQ_0
34	DSS_MCRC_DMA_REQ_1
35	FRC_EVENT_GEN_0
36	FRC_EVENT_GEN_1
37	FRC_EVENT_GEN_2
38	FRC_EVENT_GEN_3
39	RESERVED
40	LOGICAL_FRAME_START/DSS_DMMSWINT20/DSS_DMMSWINT40
41	ADC_DATA_VALID_FALL/DSS_DMMSWINT22/DSS_DMMSWINT44
42	UART_DMA_REQ_0
43	UART_DMA_REQ_1
44	DMMSW_INT_23
45	DMMSW_INT_24
46	DMMSW_INT_25
47	DMMSW_INT_26
48	DMMSW_INT_27
49	GPIO_0_host_interrupt
50	GPIO_1_host_interrupt
51	GPIO_2_host_interrupt
52	RTI1_DMA_REQ_0
53	RTI1_DMA_REQ_1
54	RTI1_DMA_REQ_2
55	RTI1_DMA_REQ_3
56	RTI2_DMA_REQ_0
57	RTI2_DMA_REQ_1
58	RTI2_DMA_REQ_2
59	RTI2_DMA_REQ_3
60	RESERVED
61	RESERVED
62	RESERVED
63	DMMSW_INT_28

3.3.14 Error Signaling Module (MSS_ESM/DSS_ESM)

The 68xx device has two instances of the Error Signaling Module (MSS_ESM/DSS_ESM), shown in Figure 3-15.

Figure 3-15. 68xx MSS_ESM/DSS_ESM Integration Diagram

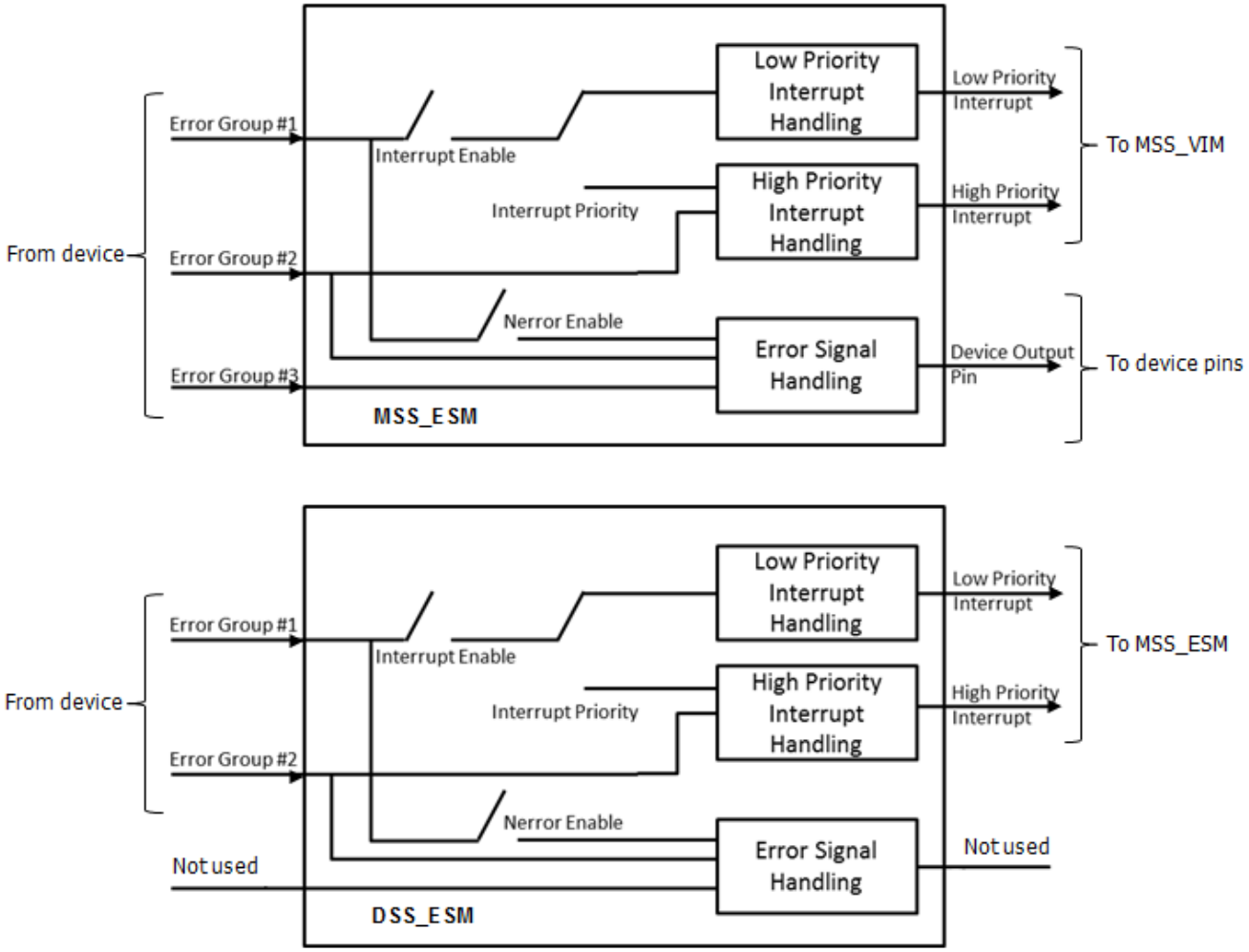


Table 3-15 shows the mapping on the input error inputs to the ESM module from various error sources available for hardware diagnostics within the device.

Table 3-15. MSS_ESM Mapping

MSS_ESM Group 1		Channel Type	Description
63	ANA_LIMP_MODE	Error Signal	Error signal at device boot-up, if the CLK monitor finds the REF CLK to be outside the permissible range of frequency
62	MSS_DCCB_ERR	Error Signal	MSS_DCCB frequency comparison error
61	MAILBOX_BSS2MSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2MSS
60	MAILBOX_BSS2MSS _REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_BSS2MSS
59	MAILBOX_MSS2BSS _FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2BSS
58	MAILBOX_MSS2BSS _REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2BSS
57	BSS_CRITICAL_ERR	Error Signal	BSS critical Error Indication. Mask control to be configured in IRQ_CTL for individual error signals.
56	Reserved	Reserved	Reserved

Table 3-15. MSS_ESM Mapping (continued)

MSS_ESM Group 1		Channel Type	Description
55	CLOCK_SUPPLY_ERR	Error Signal	Clock and Supply Errors from Analog. Mask control to be configured in IRQ_CTL for individual error signals
54	Reserved	Reserved	Reserved
53	MAILBOX_DSS2MSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2MSS
52	MAILBOX_DSS2MSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_DSS2MSS
51	MSS_MCAN_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MCAN Message RAM (MSS_MCAN/MSGMEM)
50	MSS_MCAN_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MCAN Message RAM (MSS_MCAN/MSGMEM)
49	MSS_MIBSPIB_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MIBSPIB multi-buffer (RXRAM/TXRAM)
48	MSS_MCRC_ERR	Error Signal	MSS_MCRC Comparison Error
47	MSS_MCAN_EXT_TIMESTAMP_ERR	Error Signal	MSS_MCAN Timestamping Error
46	Reserved	Reserved	Reserved
45	MSS_MIBSPIB_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MIBSPIB multi-buffer (RXRAM/TXRAM)
44	Reserved	Reserved	Reserved
43	MAILBOX_MSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2DSS
42	MAILBOX_MSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2DSS
41	DSS_ESM_GP1_ERR	Error Signal	DSS_ESM Low priority Interrupt
40:39	Reserved	Reserved	Reserved
38	DSS_CBUFF_SAFETY_ERR	Error Signal	CHIRP ERROR or CRC ERROR from DSS_CBUFF
37	DSS_ESM_GP2_ERR	Error Signal	DSS_ESM High priority Interrupt
36	DSS_TPTC1_WR_MPU_ERR	Error Signal	DSS_TPTC1 write port MPU error
35	DSS_TPTC1_RD_MPU_ERR	Error Signal	DSS_TPTC1 read port MPU error
34	HVMODE_ERR	Error Signal	Error indication from IO Supply (Supply detector for dual-voltage IOs)
33	Reserved	Reserved	Reserved
32	MSS_TCMA_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for Cortex R4F MSS_TCMA
31	Reserved	Reserved	Reserved
30	MSS_DCCA_ERR	Error Signal	MSS_DCCA frequency comparison error
29	DSS_TPTC0_WR_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
28	MSS_TCMB1_REPAIR_ERR	Alert Signal	Single-bit error/reserved indication for MSS_TCMB1
27	MSS_STC_ERR	Error Signal	MSS_STC Error indication for MSS Cortex R4F
26	MSS_TCMB0_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_TCMB0
25	MSS_MIBSPIA_MEM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_MIBSPIA multi-buffer (RXRAM/TXRAM)
24	MSS_DMA2_MEM_PARITY_ERR	Error Signal	Parity Error for MSS_DMA2 memory
23	MSS_DMA2_MPU_ERR	Error Signal	Error indication from MPU of MSS_DMA2
22	FRC_COMPARE_ERR	Error Signal	Lockstep comparison error from Free running Counter (FRC) in BSS
21	Reserved	Reserved	Reserved
20	MSS_VIM_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_VIM_RAM
19	Reserved	Reserved	Reserved
18	DSS_TPTC0_RD_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
17	MSS_MIBSPIA_MEM_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_MIBSPIA (RXRAM/TXRAM)
16	MSS_SECURE_RAM_FATAL_ERR	Error Signal	Multi-bit uncorrectable error indication for MSS_DTHE/SECURE_RAM
15	MSS_VIM_RAM_FATAL_ERR	Error Signal	Multi-bit uncorrectable error indication for MSS_VIM_RAM

Table 3-15. MSS_ESM Mapping (continued)

MSS_ESM Group 1		Channel Type	Description
14	MSS_SECURE_RAM_REPAIR_ERR	Alert Signal	Single-bit error/repair indication for MSS_DTHE/SECURE_RAM
13	Reserved	Reserved	Reserved
12	MAILBOX_BSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2DSS
11	MAILBOX_BSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_BSS2DSS
10	MAILBOX_DSS2BSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2BSS
9	DSS_CBUFF_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_CBUFF FIFO
8	DSS_CBUFF_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_CBUFF FIFO
7	DSS_TPCC_PARITY_ERR	Error Signal	Parity error from DSS_TPCC (EDMA Channel Controller)
6	NU	Reserved	Reserved
5	MAILBOX_DSS2BSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_DSS2BSS
4	MSS_CCCB_ERR	Error Signal	MSS_CCCB(Clock compare core) frequency comparison error
3	MSS_DMA_MEM_PARITY_ERR	Error Signal	Parity Error for DMA1 memory
2	MSS_DMA_MPU_ERR	Error Signal	Error indication from MPU of MSS _DMA
1	MSS_CCCA_ERR	Error Signal	MSS_CCCA(Clock compare core) frequency comparison error
0	NERROR_PAD_IN	Error Signal	Nerror from PAD looped in
MSS_ESM Group 2			
31:26	NU		
25	Reserved	Reserved	Reserved
24	MSS_RTIB_NMI	Error Signal	Watchdog Non-mask able interrupt
23:17	NU		
16	MSS_CR4F_LIVELOCK_ERR	Error Signal	Cortex R4F Live lock error
15:9	NU		
8	MSS_TCMB1_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB1
7	NU		
6	MSS_TCMB0_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMB0
5	NU		
4	MSS_TCMA_PARITY_ERR	Error Signal	Parity Error on Control signals for MSS_TCMA
3	NU		
2	Reserved	Reserved	Reserved
1:0	NU		
MSS_ESM Group 3			
31-8	NU		
7	MSS_TCMA_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMA
6	NU		
5	MSS_TCMB1_FATAL_ERR	Error Signal	Multi -bit error indication for MSS_TCMB1
4	NU		
3	MSS_TCMB0_FATAL_ERR	Error Signal	Multi-bit error indication for MSS_TCMB0
2	NU		
1	EFC_AUTOLOAD_ERR	Error Signal	Efuse Auto-load error
0	NU		

Table 3-16. DSS_ESM Mapping

DSS_ESM Group 1		Channel Type	Description
63:57	Reserved	Reserved	Reserved
56	DSS_DSP_L1P_PARITY_ERR	Error Signal	DSS_DSP_L1P Parity Error
55	MAILBOX_MSS2DSS_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from MAILBOX_MSS2DSS
54	MAILBOX_DSS2MSS_REPAIR_ERR	Error Signal	Single-bit error/repair indication from MAILBOX_DSS2MSS
53	MAILBOX_DSS2MSS_FATAL_ERR	Alert Signal	Multi-bit error indication from MAILBOX_DSS2MSS
52	MAILBOX_MSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_MSS2DSS
51	MAILBOX_BSS2DSS_REPAIR_ERR	Alert Signal	Single-bit repair indication from MAILBOX_BSS2DSS
50	MAILBOX_DSS2BSS_REPAIR_ERR	Alert Signal	Single-bit repair indication from MAILBOX_DSS2BSS
49	MAILBOX_BSS2DSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_BSS2DSS
48	MAILBOX_DSS2BSS_FATAL_ERR	Error Signal	Multi-bit error indication from MAILBOX_DSS2BSS
47	DSS_FFT_ACC_PARAM_RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from FFT PARAM
46	DSS_FFT_ACC_PARAM_RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from FFT PARAM
45	DSS_FFT_ACC_WINDOW_RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from FFT Window RAM
44	DSS_FFT_ACC_WINDOW_RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from FFT Window RAM
43	DSS_FFT_ACC_IO_RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from FFT IO RAM
42	DSS_FFT_ACC_IO_RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from FFT IO RAM
41:38	Reserved	Reserved	Reserved
37	DSS_FFT_ACC_FSM_LOCKSTEP_ERR	Error Signal	Lock step error from FFT Accelerator
36:32	Reserved	Reserved	Reserved
31	DSS_CFG_MSTID_MPU_ERR	Error Signal	Error from Master ID based MPU on the DSS_CFG_MSTID_MPU Configuration address space
30:29	Reserved	Reserved	Reserved
28	DSS_DATA_TXFR_RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_DATA_TXFR_RAM
27	DSS_DATA_TXFR_RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_DATA_TXFR_RAM
26	DSS_STC_ERR	Error Signal	Error from DSS_STC (Self-test Controller)
25	DSS_DSP_L2_UMAP_ECC_REPAIR_ERR	Alert Signal	Single bit repair indication for DSS_DSP_L2_UMAP0 or DSS_DSP_L2_UMAP1
24	DSS_HSRAM1_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_HSRAM1
23	DSS_HSRAM1_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_HSRAM1
22:19	Reserved	Reserved	Reserved
18	DSS_ADCBUF_PONG_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_ADCBUF Pong Memory
17	DSS_ADCBUF_PONG_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_ADCBUF Pong Memory
16	DSS_ADCBUF_PING_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_ADCBUF Ping Memory
15	DSS_ADCBUF_PING_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_ADCBUF Ping Memory
14	DSS_TPTC3_WR_MPU_ERR	Error Signal	DSS_TPTC3 write port MPU error
13	DSS_TPTC3_RD_MPU_ERR	Error Signal	DSS_TPTC3 read port MPU error
12	DSS_TPTC2_WR_MPU_ERR	Error Signal	DSS_TPTC2 write port MPU error
11	DSS_TPTC2_RD_MPU_ERR	Error Signal	DSS_TPTC2 read port MPU error
10	DSS_TPCC1_PARITY_ERR	Error Signal	Parity error from DSS_TPCC1 (EDMA Channel Controller)
9	DSS_CBUFF_SAFETY_ERR	Error Signal	CHIRP ERROR or CRC ERROR from DSS_CBUFF
8	DSS_TPTC1_WR_MPU_ERR	Error Signal	DSS_TPTC1 write port MPU error

Table 3-16. DSS_ESM Mapping (continued)

DSS_ESM Group 1		Channel Type	Description
7	DSS_TPTC1_RD_MPU_ERR	Error Signal	DSS_TPTC1 read port MPU error
6	DSS_TPTC0_WR_MPU_ERR	Error Signal	DSS_TPTC0 write port MPU error
5	DSS_TPTC0_RD_MPU_ERR	Error Signal	DSS_TPTC0 read port MPU error
4	DSS_CBUFF_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_CBUFF FIFO
3	DSS_CBUFF_ECC_REPAIR_ERR	Alert Signal	Single-bit error/repair indication from DSS_CBUFF FIFO
2	DSS_TPCC_PARITY_ERR	Error Signal	Parity error from DSS_TPCC (EDMA Channel Controller)
1	DSS_L3RAM_ECC_FATAL_ERR	Error Signal	Multi-bit error indication from DSS_L3RAM Memory
0	DSS_L3RAM_ECC_REPAIR_ERR	Alert Signal	Single-bit repair indication from DSS_L3RAM Memory
DSS_ESM Group 2			
31:6	Reserved	Reserved	Reserved
5	DSS_DSP_L2_UMAP_ECC_FATAL _ERR	Error Signal	Multi-bit error indication for DSS_DSP_L2_UMAP0 or DSS_DSP_L2_UMAP1
4	DSP_PBIST_ERR	Error Signal	DSP_PBIST (Memory Test) Fail Error
3	DSS_STC_ERR	Error Signal	Error from DSS_STC (Self-test Controller)
2	DSS_RTI_NMI	Error Signal	NMI from DSS_RTI (Watchdog)
1	DSS_DSP_L2_UMAP1_PARITY_ERR	Error Signal	Error from byte level parity comparison logic for DSS_DSP_L2_UMAP1
0	DSS_DSP_L2_UMAP0_PARITY_ERR	Error Signal	Error from byte level parity comparison logic for DSS_DSP_L2_UMAP0

3.3.15 High-Speed Interface (HSI)

Table 3-17 lists the high-speed interfaces available.

Table 3-17. 68xx/64xx High-Speed Interfaces

CSI2	N/A
LVDS	2 lanes

3.3.16 Handshake RAM (DSS_HSRAM1)

The 68xx device has 32KB memory for HSRAM in the DSP subsystem.

Power, Reset, Clock Management and Control Registers (IWR)

Topic **Page**

4.1	PRCM Overview	291
4.2	Digital Clock Domains	291
4.3	Resets	294
4.4	Power Domain	296
4.5	Boot	300
4.6	14xx Control Registers	301
4.7	16xx Control Registers	487
4.8	68xx/64xx Control Registers	786
4.9	18xx Control Registers	1096

4.1 PRCM Overview

PRCM manages clocks, resets, and power domain control of subsystems and modules inside the device. Additionally, configuration of certain device level features is also performed through this module. PRCM has control and status registers to achieve this functionality. The available address space of PRCM is divided as shown in Table 4-1, specific to various subsystems in the device.

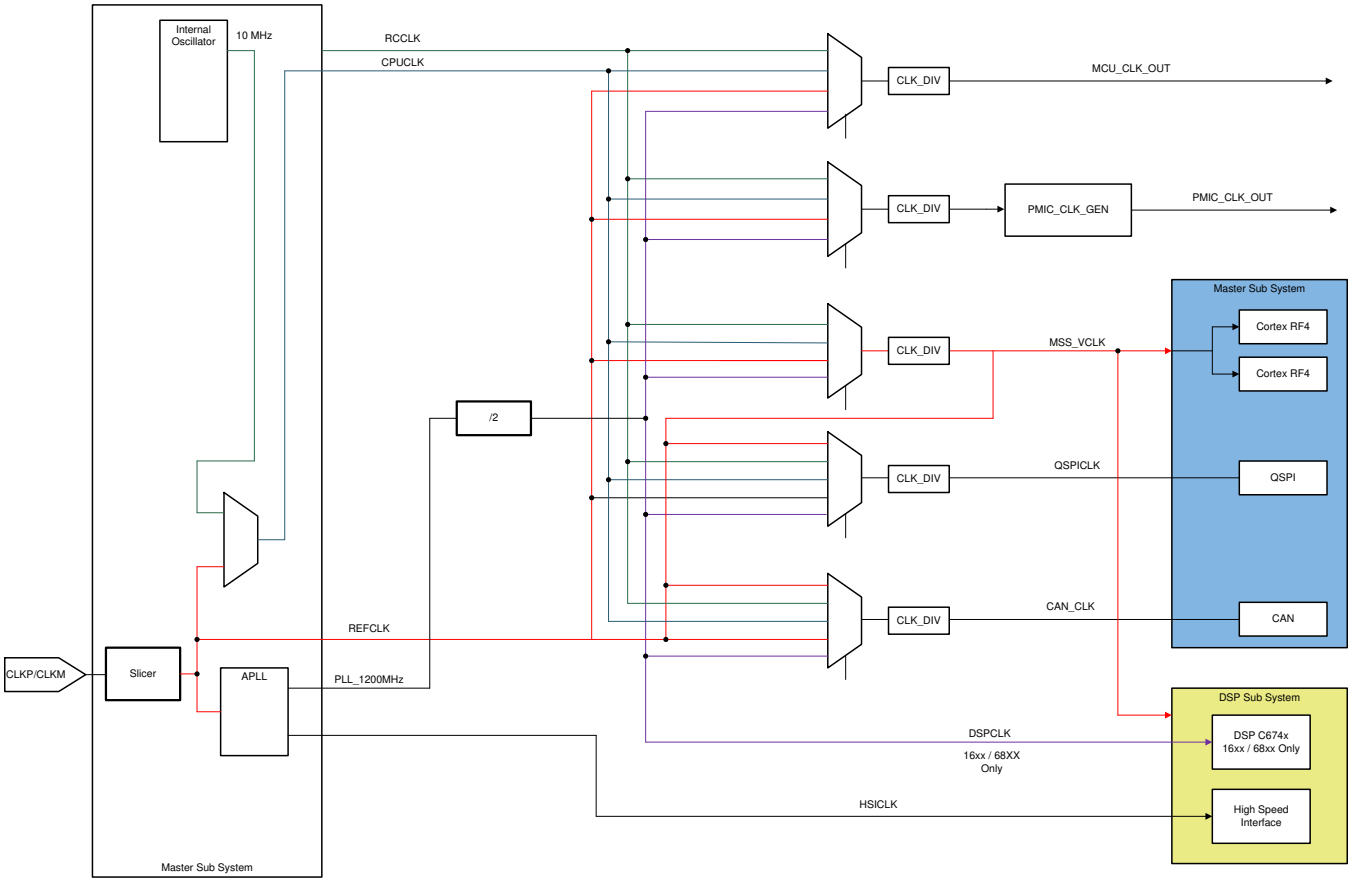
Table 4-1. PRCM Overview

PRCM Space	Description	Devices
MSS_TOPRCM	Top-level reset, clock management registers	16xx/14xx/18xx/68xx
MSS_RCM	MSS reset, clock management registers	16xx/14xx/18xx/68xx
MSS_GPCFG	General purpose control registers	16xx/14xx/18xx/68xx
DSS_REG	DSP subsystem control registers	16xx/14xx/18xx/68xx
DSS_REG2	DSPSS control module registers	16xx/18xx/68xx

4.2 Digital Clock Domains

4.2.1 Clock Overview

Figure 4-1. Clock Overview



The device has one external clock input source on the CLKP/M pins, and one internal reference clock generator. Other device-level clock sources are generated internally from these clocks by the radar subsystem. See Table 4-2 for the clocks available, based on the device.

4.2.1.1 REFCLK

External clock input can be a forced clock, or generated through a crystal between the CLKP/M pins. REFCLK is expected to operate between 40 MHz and 100 MHz. See Table 4-2 for a set of supported frequencies.

4.2.1.2 RCCLK

The RCCLK is generated by internal oscillator and is 10 MHz +/- 10%.

4.2.1.3 CPUCLK

If the REFCLK is within the permissible range of frequency of 40 MHz to 100 MHz, the CPUCLK is fed through a version of REFCLK. If the REFCLK is not within the range, the CPUCLK is switched to RCCLK.

4.2.1.4 PLLCLK

REFCLK is used as reference for PLL in the design to generate 1200-MHz clock, and its divide down versions as listed in Table 4-2.

4.2.1.5 HSICLK

This is a 300-MHZ to 1800-MHz clock generated by PLL for high-speed interfaces, such as LVDS and CSI.

4.2.2 Device Clock Sources (For Digital Logic)

Table 4-2 shows a list of available clock sources in the device and their description. These clocks serve as source clocks for various clock domains in the device. Configuration of these clocks is performed by boot ROM code.

Table 4-2. Clock Sources

Clock Source Name	Frequency	Description	Devices
RCCLK	10 MHz +/- 5%	Output of internal oscillator	16xx/14xx/18xx/68xx
REFCLK	40 MHz, 50 MHz, 80 MHz, 100 MHz	Same as the clock at input pins CLKP/M slicer. This pin can be either driven by an external clock input or connected to a crystal.	16xx/14xx/18xx/68xx
CPUCLK	RCCLK or REFCLK	The CPU CLK is fed through of the REF CLK once the CLK monitor ensures the REF CLK is within the permissible range of frequency (40 to 100 MHz). In the event the REF CLK is not within the range, the CPU CLK is switched to RCCLK during the course of wake-up.	16xx/14xx/18xx/68xx
PLLCLK_1200	1200 MHz	Output of internal PLL. REFCLK serves as the reference input to this PLL.	16xx/14xx/18xx/68xx
PLLCLK_600	600 MHz	Divided version of PLLCLK_1200 clock	16xx/14xx/18xx/68xx
HSICLK	300 to 1800 MHz	High speed clock used by LVDS/CSI interface. Programmed and controlled using the Radar Subsystem API.	16xx/14xx/18xx/68xx

4.2.3 Device Clock Domains

Table 4-3 shows the clock domains available in the device that are generated from available clock sources.

Table 4-3. Clock Domains

Clock Domain	Selectable Clock Sources	Clock Source Select (Module.Register.Field)	Clock Divider (Register.Field)	Clock Gate Register (Register.Field)	Frequency	Description	Devices
MSS_VCLK	CPUCLK RCCLK REFCLK PLLCLK_600	MSS_RCM.CLKSRCSEL1.VCLKCLKSRCSEL	MSS_RCM.CLKDIVCTL0.VCLKCLKDIV	NA	200 MHz maximum	Clock for Cortex CR4F and other modules with following exceptions: DSP Core, QSPI, CAN	16xx/14xx/18xx/68xx
DSPCLK	PLLCLK_600	NA	NA	NA	600 MHz maximum	Clock source for DSP sub system. Fixed value.	16xx/18xx/68xx
QSPICLK	CPUCLK RCCLK REFCLK PLLCLK_600 MSS_VCLK	MSS_RCM.CLKSRCSEL0.QSPICLKSRCSEL	MSS_RCM.CLKDIVCTL2.QSPICLKDIV	MSS_RCM.CLKGATE.QSPICLKGATE	60 MHz maximum	QSPI peripheral clock	16xx/14xx/18xx/68xx
CAN_CLK	CPUCLK RCCLK REFCLK PLLCLK_600 MSS_VCLK	MSS_RCM.CLKSRCSEL0 DCANCLKSRCSEL	MSS_RCM.CLKDIVCTL0 DCANCLKDIV	MSS_RCM.CLKGATE.DCANCLKGATE	10 MHz maximum	CAN peripheral clock	16xx/18xx/68xx
MCU_CLK	CPUCLK RCCLK REFCLK PLLCLK_600	TOP_RCM.EXTCLKSRCSEL.EXTCLK1SRCSEL	TOP_RCM.EXTCLKDIV.EXTCLK1DIV	MSS_TOPRCM.EXTCTL.EXTCLK1GATE	20 to 25 MHz default. 15 - 40 MHz configurable	Clock out from device. Intended for external microcontroller	16xx/14xx/18xx/68xx
PMIC_CLK	CPUCLK RCCLK REFCLK PLLCLK_600	TOP_RCM.EXTCLKSRCSEL.EXTCLK2SRCSEL	TOP_RCM.EXTCLKDIV.EXTCLK2DIV	MSS_TOPRCM.EXTCTL.EXTCLK2GATE	1.7 to 2.2 MHz configurable	Clock out from device. Intended for external PMIC	16xx/14xx/18xx/68xx

Configuration of the clock domains in terms of clock source selection and frequency is controlled through registers mentioned in Table 4-3. Multiple clock sources are available for each of the clock domains; required clock sources can be selected by the clock domains clock source register. Clock domains have programmable divider registers which allow the clock domain to get a divided down clock of the selected source clock.

Boot ROM code performs the default initialization of these registers. MSS_VCLK is setup before other clocks; this is the clock for most of the device components and subsystems except for the peripherals such as those mentioned in Table 4-3, which have their independently selectable clock sources and divider values.

4.2.4 Clock Initialization Sequence

Typically the clock initialization of MSS_VCLK and DSPCLK is managed by boot ROM, and the user is not expected to alter these configurations.

The initialization sequence for QSPICLK, CAN_CLK, MCU_CLK, and PMIC_CLK is listed below. See the applicable register.field for the clock in Table 4-3.

1. Disable the clock by gating off the clock using the Clock Gate register field.
2. Write a divider value based on the desired clock frequency to the clock divider register field.
3. Write the clock source register field with the selection from the available clock sources mentioned in Table 4-2. The corresponding clock source values are mentioned in the register field descriptions.
4. Enable the clock by gating the clock by writing to the Clock Gate register field.

4.2.5 PMIC Clock

The generated PMIC_CLK can be divided down further in the PMIC_CLK_GEN module by configuring the registers shown in Table 4-4. These registers should be programmed before doing the setup of PMIC_CLK registers listed in Table 4-3.

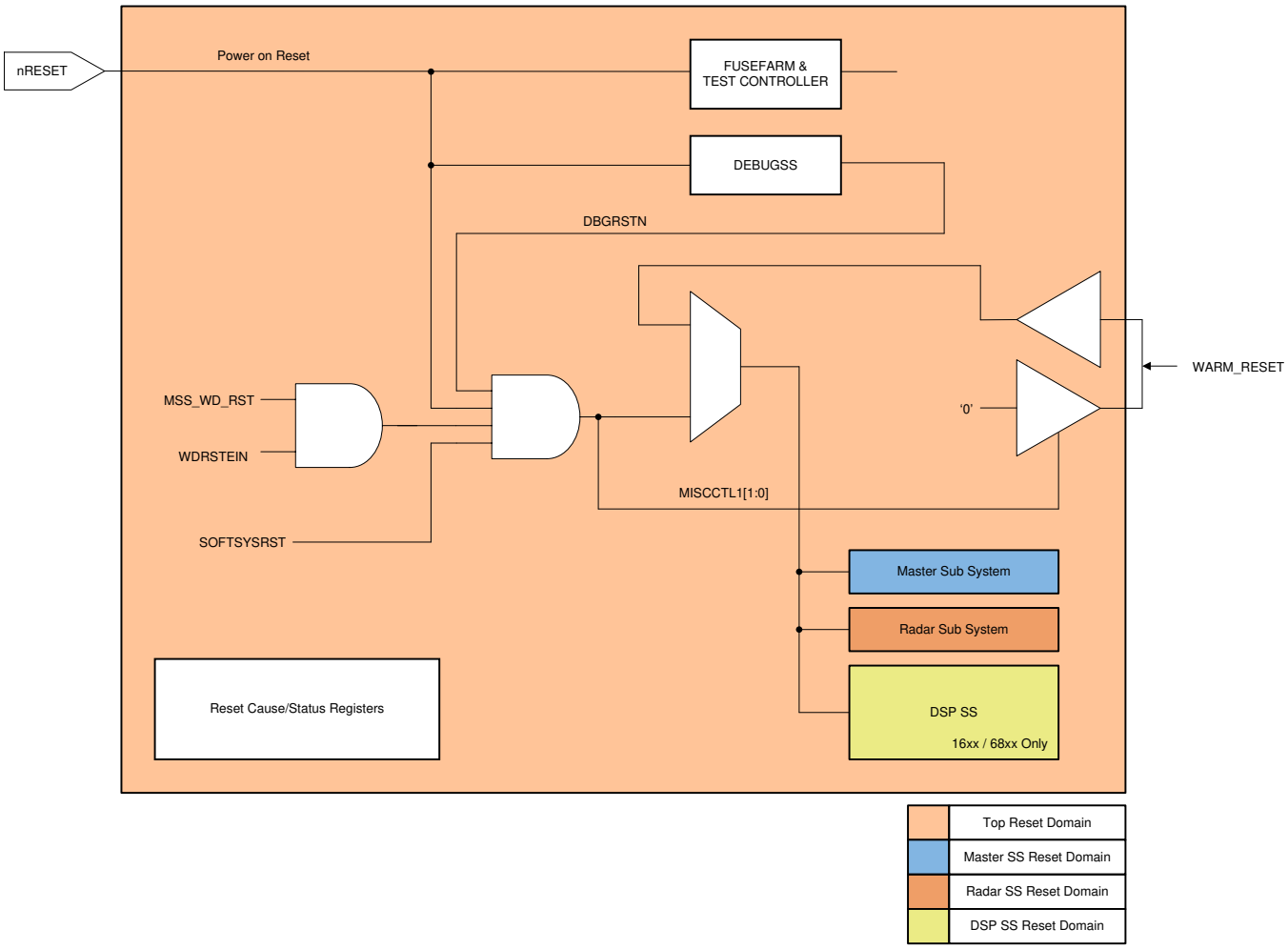
Table 4-4. PMIC Registers

Register	Description
MSS_TOPRCM.DCDCCTL0.DCDCCTL0	Controls the slope of the output PMIC_CLK_OUT. Set to 0.
MSS_TOPRCM.DCDCCTL1.DCDCRST	Dithering control block reset. Set to 0
MSS_TOPRCM.DCDCCTL1.DCDCLKEN	Enable Clk. Set to 1.
MSS_TOPRCM.DCDCCTL1.DCDCCTL1[8]	Set to 0.
MSS_TOPRCM.DCDCCTL1.DCDCCTL1[9]	Set to 0.
MSS_TOPRCM.DCDCCTL1.DCDCCTL1[23:16]	Set to divider value to get appropriate PMIC clk out frequency. Minimum 2.
MSS_TOPRCM.DCDCCTL1.DCDCCTL1[31:24]	Set to divider value to get appropriate PMIC clk out frequency. Minimum 2.

4.3 Resets

4.3.1 Reset Overview

Figure 4-2. Reset Overview



Two device resets are available that can be controlled from the device pins: the power-on reset pin NRESET and the bidirectional WARM_RESET signal. The warm reset signal is implemented as an I/O, so that an external monitor can be utilized to detect changes to the state of the internal warm reset control signal.

Device registers can capture recent reset events, and can be used by software to manage failure recovery. Certain registers in the device are immune to WARM_RESET, and can only be reset by power-on reset.

4.3.2 Reset Types and Sources

Table 4-5. Reset Types and Sources

Reset Type	Reset Source	Description	Devices
Power On Reset	Device Pin NRESET	Reset triggered by the device reset pin NREST. This resets the entire device, including all subsystems and interfaces. This is an active low asynchronous Power ON reset signal, and must be asserted for minimum 500 ns to reset the device.	16xx/14xx/18xx/68xx
Warm Reset	Soft Reset, Watch Dog Reset, Device Pin WARM_RESET pin	This is an active low warm reset internally generated by the device, or triggered by device pin WARM_RESET. A write to the TOP_RCM.SOFTSYSRST register or watch dog module can generate this reset; additionally, the external pin WARM_RESET can also be used to trigger this reset. Selection is controlled through the TOP_RCM.MISCTL1 register. The WARM_RESET pin is an open-drain failsafe IO which can be used to reset the device from the external world or to report the reset to the external world if it is generated by an internal source such as watchdog. A write to the TOP_RCM.SOFTSYSRST, as explained in register description, can create this reset. The watch dog module in the master subsystem can be configured to trigger this reset. Check the watch dog module description for details. This feature is enabled by the TOP_RCM. WDRSTEN register.	16xx/14xx/18xx/68xx

As listed in Table 4-5, various reset sources can generate the different resets used inside the device to reset various components and submodules.

A Warm_rest excludes certain modules and spaces listed in Table 4-5 from reset. These can only be reset by power-on reset.

4.3.3 Reset Domains

The device can be divided into various reset domains, as listed in Table 4-6. The top reset domains cover the entire device and all of the subsystems. Additional subsystem-level reset domains are available, and can be reset independently based on resets mentioned in Table 4-6.

Table 4-6. Reset Domains

Reset Domain	Description	Resets	Devices
Top reset domain	This top device-level reset domain resets entire device and all subsystems. All other reset domains are subdomains of this domain, and resetting this domain issues a reset to these subdomains. Only power-on reset can reset this domain, and it is immune to any other system reset type. Sub reset domain can be independently reset, as mentioned in the respective rows.	Power-on reset	16xx/14xx/18xx/68xx
Master subsystem reset domain	This reset subdomain controls the reset to the master susbsystem and the modules inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset	16xx/14xx/18xx/68xx
Radar subsystem reset domain	This reset subdomain controls the reset to the radar susbsystem and all components inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset	16xx/14xx/18xx/68xx

Table 4-6. Reset Domains (continued)

Reset Domain	Description	Resets	Devices
DSP subsystem reset domain	This reset subdomain controls the reset to the DSP subsystem and all components inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset	16xx/14xx/18xx/68xx

4.3.4 Reset Cause Registers

Various reset cause registers are available that show the last reset issued to the device or subsystem, as shown in Table 4-7.

Table 4-7. Reset Cause Registers

Register	Description
MSS_RCM.RSTCAUSE.RSTCAUSE	Records the cause for the master SS reset domain reset
MSS_TOPRCM.SYSRSTCAUSE.SYSRSTCAUSE	Records the cause for the top reset domain reset
DSS_REG.GEMRSTCAUSE.GEMLRSTCAUSE	Records the cause of the local reset of the DSP reset domain
DSS_REG.GEMRSTCAUSE.GEMGRSTCAUSE	Records the cause of the global reset of the DSP reset domain
DSS_REG.GEMRSTCAUSE.GEMPORCAUSE	Records the cause of the POR reset of the DSP reset domain

4.3.5 Module Resets

4.3.5.1 MSS CR4 Soft Reset Sequence

Use the steps below to soft reset the MSS CR4:

1. Write 0xAD to ALTIUS:MSS_RCM:SOFTCORERST:RST_WFICHECKEN (before asserting the reset to CR4, wait for CR4 to enter WFI).
2. Write 0xAD to ALTIUS:MSS_RCM:SOFTRST1:CR4SYSRST (write 0xAD to assert a MSS CR4-only reset).

This automatically asserts and deasserts the MSS CR4 reset.

4.3.5.2 VIM Reset Consideration

Issue a soft reset to the VIM module using the following steps:

1. Write value 0xAD to register MSS_RCM:SOFTRST2:VIMRST.
2. Release reset for VIM module by writing 0x0 to register MSS_RCM:SOFTRST2:VIMRST. This only resets the VIM module and its state.

NOTE: This does not reset the RAM memory within the VIM module.

4.4 Power Domain

Switchable power domains are only for the 16xx/18xx/68xx device.

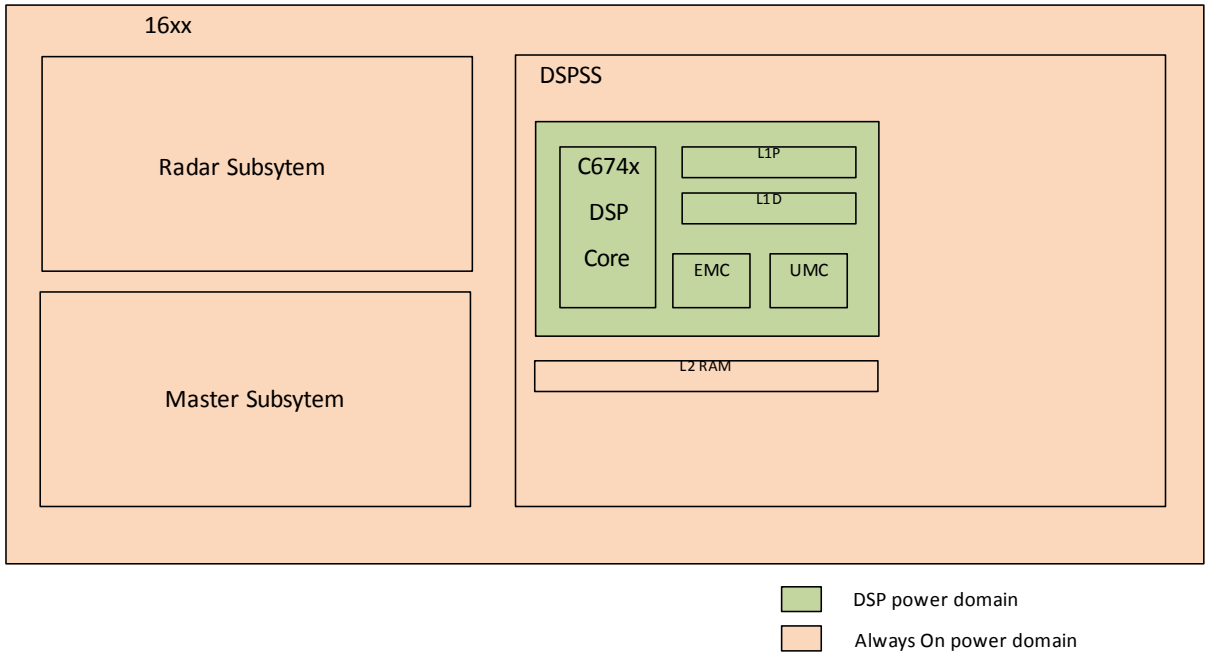
4.4.1 Available Device Power Domains

The device has two power domains:

- Always-on power domain: This power domain supplies power to entire device except for the DSP core. It cannot be switched off dynamically within the device.
- DSP power domain: The DSP C674x core inside the DSP subsystem is placed on switchable power

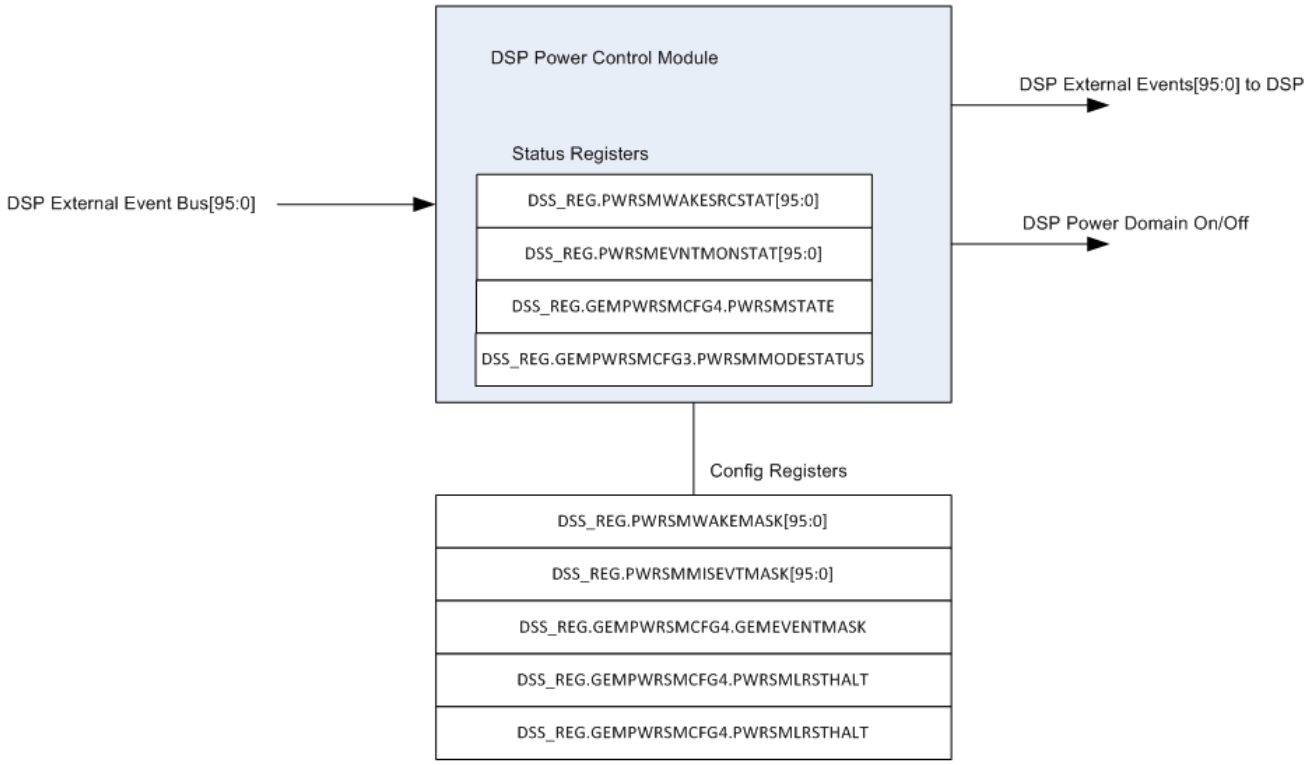
domain, and can be switched on or off as needed. This feature can be used to reduce power consumption of the device when the DSP core is not in use. By default, the state of this power domain is OFF.

Figure 4-3. Power Domains



4.4.2 Power Domain Control

Figure 4-4. Power Domain Controls



The DSP can be power cycled ON and OFF using the DSP power control module to save power when inactive. On POR, the DSP is powered OFF.

Any of the external DSP events can be used as a wakeup event to power on the DSP. The events can be configured as wakeup events by unmasking the event in DSS_REG.PWRSMWAKEMASK. After power up, the events that triggered the wakeup are registered in DSS_REG.PWRSMWAKESRCSTAT.

During power OFF, the control module ensures that none of the external DSP events are sent to the DSP. This is done by setting the DSS_REG.GEMPWRSMCFG4.GEMEVENTMASK field before triggering a power down. When this field is set, the events are captured inside the control module, so there is a record of the missed events during power-off. When the DSP is powered back on, the SW clears the DSS_REG.GEMPWRSMCFG4.GEMEVENTMASK field so that any new HW events are now routed to the DSP. Any events that occurred during power off are saved in the PWRSMEVNTMONSTAT register. There is an option provided to send any of the missed events directly on the HW lines to the DSP at the time of clearing the DSS_REG.GEMPWRSMCFG4.GEMEVENTMASK after power up. To enable this feature, unmask these events in the PWRSMMISEVTMASK registers to enable any missed event to propagate on the HW lines on clearing the DSS_REG.GEMPWRSMCFG4.GEMEVENTMASK field. After processing the missed events and wakeup source, the SW must clear all these events by writing to the DSS_REG.PWRSMWAKESRCSTATCLR register.

4.4.2.1 Power Domain Status

The current status of the DSP power domain can be read at DSS_REG.GEMPWRSMCFG3.PWRSMMODESTATUS. Table 4-8 shows the decoding of the value read.

Table 4-8. Power Domain Status

PWRSMMODESTATUS	DSP Power Domain State
0	GEM is OFF
1	GEM transitioning from OFF to ON
2	GEM transitioning from ON to OFF
3	GEM is ON

4.4.2.2 Power ON Control Sequence

Below is the detailed sequence to power on the DSP.

Master subsystem-controlled DSP power ON sequence — This sequence is used to power on the DSP core.

1. Ensure the DSP resets from the TOP RCM are de-asserted. The resets are then fully controlled by the power state machine.
2. Unmask the wakeup event: by default, all DSP interrupts are masked as wakeup events. To unmask a wakeup event, write the corresponding bit in registers DSS_REG.PWRSMWAKEMASK[0-2]. The 96-bit configuration corresponds to the [95:0] interrupt lines to the DSP.
3. If the program code must be downloaded into the DSP L2 memory before starting the DSP execution, then set the DSS_REG.GEMPWRSMCFG4.PWRSMLRSTHALT bit to 0x1.
4. Trigger the wakeup: when an unmasked interrupt is generated to the DSP, a wakeup trigger is generated.
5. Wait for DSP to be powered on by polling the field DSS_REG.GEMPWRSMCFG4.PWRSMSTATE for the value 0xB. SW can now configure and trigger the DMA for the program code download.
6. When the program code is completed, SW can now release the halt by clearing the DSS_REG.GEMPWRSMCFG4.PWRSMLRSTHALT bit; DSP begins execution.
7. Follow the steps in the DSP Execution on Power ON section below.

Autonomous DSP power ON HW event — This sequence is used to power on the DSP after a DSP power off sequence where the event used to wakeup is already unmasked before power down in the DSS_REG.PWRSMWAKEMASK[0-2].

1. The unmasked HW event triggers the DSP wakeup and power on DSP, and DSP begins execution.
2. In the DSP program execution, after booting, unmask all the events to the DSP by clearing the DSS_REG.GEMPWRSMCFG4.GEMEVENTMASK field to 0x0.
3. Any events that occurred between the previous GEM_EVENT_MASK set and unmasked in DSS_REG. DSS_REG__PWRSMMISEVTMASK[0-2] appear on the HW interrupt request lines to be serviced.
4. Reading the DSS_REG.PWRSMWAKESRCSTAT[0-2] provides the event that caused the wakeup.
5. Reading the DSS_REG.PWRSMEVNTMONSTAT[0-2] provides the events that occurred between the previous GEM_EVENT_MASK set.
6. Writing 0x1 to the field in the DSS_REG.PWRSMWAKESRCSTATCLR[0-2] clears the corresponding interrupt[95:0] status in the DSS_REG.PWRSMWAKESRCSTAT and DSS_REG.PWRSMEVNTMONSTAT

4.4.2.3 Power OFF Control Sequence

Below is the detailed sequence to power off the DSP.

DSP execution to power OFF the DSP:

1. Ensure that all previous registered events during the previous power down are cleared.
2. Set up the DSP interrupt routine for INTH_INT_ID_DSS_PDC_INT (Gem event number : 118)

a. This interrupt routine executes when the Clock Stop request is sent to the DSP, such that DSP sends back the Clock Stop Ack Set the bit DSP_ICFG.PDCCMD.GEMPD and executes the "idle" instruction.

3. If GEM powers up in autonomous mode without the program download, set the DSS_REG.GEMPWRSMCFG4.PWRSMLRSTHALT bit to 0x0.
4. Unmask the events to use as wakeup events in DSS_REG.PWRSMWAKEMASK[0-2].
5. Unmask the events to receive as HW event pulses after the next power up in DSS_REG. DSS_REG__PWRSMMISEVTMASK[0-2].
6. Mask all interrupts from DSP by setting the DSS_REG.GEMPWRSMCFG4.GEMEVENTMASK field to 0x1.
7. Trigger a power down by setting the DSS_REG.GEMPWRSMCFG4.PWRSMSLEEPTRIG field to 0x1, and wait in a while loop. This triggers the interrupt INTH_INT_ID_DSS_PDC_INT, followed by the power down of the DSP core by the control module.

The state of the DSP power domain can be polled by other master subsystem CR4F cores to ensure the DSP power down.

4.5 Boot

4.5.1 Bootloader Modes

The device supports two boot modes, Functional Mode and Flashing Mode. During boot, the device checks the value on the device pins SOP0, SOP1, and SOP2, to determine the boot mode. After boot, any change in the value does not affect the boot mode of the device. To change the boot mode, the new SOP values must be applied before power-cycling the device. Refer to Table 4-9 for the SOP pin values for each mode.

Table 4-9. SOP Table

SOP2	SOP1	SOP0	Bootloader mode and operation
0	0	1	Functional Mode This mode is used to read the image from the onboard serial flash. The image is loaded into the internal RAM and executes it.
1	0	1	Flashing Mode This mode is used to write the image to the onboard serial flash. The image is accepted over UART, and loads it to the serial flash connected to the QSPI port.

NOTE: These are the only valid SOP combinations supported by the device. Any other values may result in unknown device behavior.

4.5.2 ROM Eclipsing

ROM eclipsing is a feature that is used to change the visibility of MSS_TCMA_ROM in the master subsystem. Typically at address 0x0000_0000, MSS_TCMA_ROM address space begins. However, when ROM eclipsing is enabled, the address 0x0000_0000 gets remapped to the start address MSS_TCMA_RAM. As a result, MSS_TCMA_ROM is effectively eclipsed and not accessible in the address space.

After the device boots, the application view is always the post-eclipse view, as the ROM eclipsing step is performed by the device bootrom code. The pre-eclipse view is provided for completeness purposes. Refer to the device-specific memory map for unlisted modules.

Table 4-10. Pre-Eclipse View of Master Subsystem Cortex-R4F

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_TCMA_ROM	0x0000_0000	0x0001_7FFF	128KiB	MSS_TCMA_ROM (TCMA) Program ROM

Table 4-10. Pre-Eclipse View of Master Subsystem Cortex-R4F (continued)

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
Reserved	0x0001_8000	0x001F_FFFF		Reserved
MSS_TCMA_RAM	0x0020_0000	0x07FF_FFFF	128KiB/256KiB/ 512KiB	MSS_TCMA_RAM (TCMA) size varies based on device and DSS_L3 (L3) sharing options configured
MSS_TCMB	0x0800_0000	0x0C1F_FFFF	192KiB	MSS_TCMB (TCMB) Data RAM

Table 4-11. Post-Eclipse View of Master Subsystem Cortex-R4F

Module Name	Frame Address (Hex)		Size Used	Description
	Start	End		
MSS_TCMA_RAM	0x0000_0000	0x07FF_FFFF	128KiB/256KiB/5 12KiB	MSS_TCMA_RAM (TCMA) size varies based on device and DSS_L3 (L3) sharing options configured
MSS_TCMB	0x0800_0000	0x0C1F_FFFF	192KiB	MSS_TCMB (TCMB) Data RAM

Table 4-12 lists the steps to enable the ROM eclipsing feature, as performed by the device bootrom (the application does not need to do this).

Table 4-12. Enable ROM Eclipsing

#	Step	Description
1	Write 0xAD to MSS_RCM.CR4CTL.MEMSWAPWAIT	Post-eclipse address map is effective only after soft reset is issued to Cortex R4F.
2	Write 0xAD to MSS_RCM.CR4CTL.CR4MEMSWAP	Enables the eclipsing feature that allows the address remap. The Cortex R4F does not see the new address space until it goes through a soft reset.
3	Write 0xAD to MSS_RCM.SOFTRST1.CR4SYSRST	Issue soft reset to Cortex R4F. This makes the post-eclipse address space visible.

4.6 14xx Control Registers

Registers from the TOP_RCM, MSS_RCM, MSS_GPCFG, DSS_REG, and DSS_REG2 provide control and status information over various features of the SoC. The following sections provide a description of the registers and fields available under these modules specific to the 14xx and 16xx/18xx/68xx devices.

4.6.1 MSS_TOPRCM Registers

Table 4-802 lists the memory-mapped registers for the MSS_TOPRCM. All register offset addresses not listed in Table 4-802 should be considered as reserved locations and the register contents should not be modified.

Table 4-13. MSS_TOPRCM Registers

Offset	Acronym	Register Name	Section
8h	BSSCTL	Control Signals to BSS	Section 4.6.1.1
10h	EXTCLKDIV	Clock divide value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.6.1.2
14h	EXTCLKSRCSEL	Clock source select value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.6.1.3
18h	EXTCLKCTL	Clock gate control for MCU_CLKOUT and PMIC_CLKOUT	Section 4.6.1.4
1Ch	SOFTSYSRST	Software triggered Warm Reset	Section 4.6.1.5
20h	WDRSTEN	Issue Warm reset upon MSS Watch dog reset	Section 4.6.1.6
24h	SYSRSTCAUSE	Reset cause register	Section 4.6.1.7
28h	SYSRSTCAUSECLR	Clear Reset Cause register	Section 4.6.1.8
34h	MISCCAPT	Capture required Status values across the chip.	Section 4.6.1.9
38h	DCDCCTL0	PMIC_CLKOUT dethering control	Section 4.6.1.10
3Ch	DCDCCTL1	PMIC_CLKOUT dethering control	Section 4.6.1.11
44h	MISCCTL	Miscellaneous Control register	Section 4.6.1.12
48h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.6.1.13
4Ch	LVDSPADCTL0	LVDS pad control	Section 4.6.1.14
50h	LVDSPADCTL1	LVDS pad control	Section 4.6.1.15
60h	DFTREG0		Section 4.6.1.16
64h	DFTREG1		Section 4.6.1.17
68h	DFTREG2		Section 4.6.1.18
6Ch	DFTREG3		Section 4.6.1.19
70h	DFTREG4		Section 4.6.1.20
74h	DFTREG5		Section 4.6.1.21
78h	SHMEMINITADDR	Start and End Address for Shared Memory Initialization	Section 4.6.1.22
7Ch	SHMEMINITECC	ECC value to be written into Shared Memory during initialization.	Section 4.6.1.23
A0h	DSSMEMBANKEN	Shared Memory Bank Enable	Section 4.6.1.24
A4h	DSSMEMTAB0	Shared Memory Table	Section 4.6.1.25
A8h	DSSMEMTAB1	Shared Memory Table	Section 4.6.1.26
B0h	TCMAMEMBANK_EN	Shared Memory Bank Enable	Section 4.6.1.27
B4h	TCMAMEMTAB0	Shared Memory Table	Section 4.6.1.28
B8h	TCMAMEMTAB1	Shared Memory Table	Section 4.6.1.29
C0h	TCMBMEMBANKEN	Shared Memory Bank Enable	Section 4.6.1.30
C4h	TCMBMEMTAB0	Shared Memory Table	Section 4.6.1.31
C8h	TCMBMEM_TAB1	Shared Memory Table	Section 4.6.1.32
D8h	MEMINITSTART	Memory Initialization trigger bit for memories in BSS and DSS	Section 4.6.1.33
DCh	MEMINITDONE	Memory Initialization done status for memories in BSS and DSS	Section 4.6.1.34
FCh	MSS_SIGNATURE	Spare Register	Section 4.6.1.35
178h	MISCCTL1	Miscellaneous Control Register	Section 4.6.1.36
180h	USERMODEEN2	USER_MODE_ACCESS_EN	Section 4.6.1.37
18Ch	SYSTICK		Section 4.6.1.38

Complex bit access types are encoded to fit into small table cells. Table 4-803 shows the codes that are used for access types in this section.

Table 4-14. MSS_TOPRCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.6.1.1 BSSCTL Register (Offset = 8h) [reset = ADADADADh]

BSSCTL is shown in Figure 4-767 and described in Table 4-804.

Return to Summary Table.

Control Signals to BSS

Figure 4-5. BSSCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BSSCPUHALT								RESERVED							
R/W-ADh								R/W-ADh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
R/W-ADh															

Table 4-15. BSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BSSCPUHALT	R/W	ADh	Halt BSS CR4, To halt : either 3:0 should be 0xD or 7:4 should be 0xA One should Halt the processor before releasing BSS reset.
23-0	RESERVED	R/W	ADh	Reserved

4.6.1.2 EXTCLKDIV Register (Offset = 10h) [reset = 0h]

EXTCLKDIV is shown in Figure 4-769 and described in Table 4-806.

Return to Summary Table.

Clock divide value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-6. EXTCLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2DIV								EXTCLK1DIV							
R/W-0h								R/W-0h							

Table 4-16. EXTCLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU			Reserved
15-8	EXTCLK2DIV	R/W	0h	Divide value for PMIC_CLKOUT source clock selected by field EXTCLKSRCSEL in register CLKSRCSEL. “0000_0000” => div1 “0000_0001” => div2 ... “1111_1111” => div255 Change the divide value before switching to the new clock. Switching to the new clock is done by programming EXTCLK2SRCSEL.
7-0	EXTCLK1DIV	R/W	0h	Divide value for MCU_CLKOUT (the one going out of chip) source clock selected by field EXTCLKSRCSEL in register CLKSRCSEL. “0000_0000” => div1 “0000_0001” => div2 ... “1111_1111” => div255 Change the divide value before switching to the new clock. Switching to the new clock is done by programming EXTCLK1SRCSEL.

4.6.1.3 EXTCLKSRCSEL Register (Offset = 14h) [reset = 0h]

EXTCLKSRCSEL is shown in Figure 4-770 and described in Table 4-807.

Return to Summary Table.

Clock source select value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-7. EXTCLKSRCSEL Register

31	30	29	28	27	26	25	24
NU1							
23	22	21	20	19	18	17	16
NU1							
15	14	13	12	11	10	9	8
NU1				EXTCLK2SRCSEL			
				R/W-0h			
7	6	5	4	3	2	1	0
NU0				EXTCLK1SRCSEL			
				R/W-0h			

Table 4-17. EXTCLKSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU1			Reserved
11-8	EXTCLK2SRCSEL	R/W	0h	Select clock source for PMIC_CLKOUT 2:0 – BISTSS Clock source in use. 000 => CPUCLK 001, 100, 101, 111 => RCCLK 010 => 600-Mhz PLL divided clock 110 => REFCLK from ANA
7-4	NU0			Reserved
3-0	EXTCLK1SRCSEL	R/W	0h	Select clock source for MCU_CLKOUT 2:0 – BISTSS Clock source in use. 000 => CPUCLK 001, 100, 101, 111 => RCCLK 010 => 600-Mhz PLL divided clock 110 => REFCLK from ANA

4.6.1.4 EXTCLKCTL Register (Offset = 18h) [reset = ADADh]

EXTCLKCTL is shown in Figure 4-771 and described in Table 4-808.

Return to Summary Table.

Clock gate control for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-8. EXTCLKCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2GATE								EXTCLK1GATE							
R/W-ADh								R/W-ADh							

Table 4-18. EXTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU			Reserved
15-8	EXTCLK2GATE	R/W	ADh	Pre clock divider gate for PMIC_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA
7-0	EXTCLK1GATE	R/W	ADh	Pre clock divider gate for MCU_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA

4.6.1.5 SOFTSYSRST Register (Offset = 1Ch) [reset = 0h]

SOFTSYSRST is shown in Figure 4-772 and described in Table 4-809.

Return to Summary Table.

Software triggered Warm Reset

Figure 4-9. SOFTSYSRST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SOFTSYSRST							
								R/W-0h							

Table 4-19. SOFTSYSRST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU			Reserved
7-0	SOFTSYSRST	R/W	0h	Write 0xAD to trigger warm reset to the chip.

4.6.1.6 WDRSTEN Register (Offset = 20h) [reset = 0h]

WDRSTEN is shown in Figure 4-773 and described in Table 4-810.

Return to Summary Table.

Issue Warm reset upon MSS Watch dog reset

Figure 4-10. WDRSTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								WDRSTEN							
																								R/W-0h							

Table 4-20. WDRSTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU			Reserved
7-0	WDRSTEN	R/W	0h	Write 0xAD to trigger warm reset to the chip upon MSS wdog reset.

4.6.1.7 SYSRSTCAUSE Register (Offset = 24h) [reset = 0h]

SYSRSTCAUSE is shown in Figure 4-774 and described in Table 4-811.

Return to Summary Table.

Reset cause register

Figure 4-11. SYSRSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU												SYSRSTCAUSE			
												R-0h			

Table 4-21. SYSRSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU			Reserved
3-0	SYSRSTCAUSE	R	0h	Gives cause of chip reset “1001” : System out of NRESET “1000” : External Warm Reset “1010” : Warm reset because of MSS Wdog “1100” : Warm reset because of Software trigger. Note: The ROM Bootloader clears this register as part of its processing, so it always reads zero by the application or secondary bootloader. However, the ROM Bootloader saves the original value into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.6.1.8 SYSRSTCAUSECLR Register (Offset = 28h) [reset = 0h]

SYSRSTCAUSECLR is shown in Figure 4-775 and described in Table 4-812.

Return to Summary Table.

Clear Reset Cause register

Figure 4-12. SYSRSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SYSRSTCAUSECLR							
0h								0h							

Table 4-22. SYSRSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU			Reserved
7-0	SYSRSTCAUSECLR		0h	Write 0xAD to clear SYSRSTCAUSE.

4.6.1.9 MISCCAPT Register (Offset = 34h) [reset = 0h]

MISCCAPT is shown in Figure 4-776 and described in Table 4-813.

Return to Summary Table.

Capture required Status values across the chip.

Figure 4-13. MISCCAPT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCAPT																															
R-0h																															

Table 4-23. MISCCAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCAPT	R	0h	0: No error Any other non-zero value: Error

4.6.1.10 DCDCCTL0 Register (Offset = 38h) [reset = 0h]

DCDCCTL0 is shown in Figure 4-777 and described in Table 4-814.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-14. DCDCCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCDCCTL0																															
R/W-0h																															

Table 4-24. DCDCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DCDCCTL0	R/W	0h	PMIC_CLKOUT dethering control

4.6.1.11 DCDCCTL1 Register (Offset = 3Ch) [reset = 0h]

DCDCCTL1 is shown in Figure 4-778 and described in Table 4-815.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-15. DCDCCTL1 Register

31	30	29	28	27	26	25	24
DCDCCTL1							
R/W-0h							
23	22	21	20	19	18	17	16
DCDCCTL1							
R/W-0h							
15	14	13	12	11	10	9	8
DCDCCTL1							
R/W-0h							
7	6	5	4	3	2	1	0
DCDCCTL1						DCDCLKEN	DCDCRST
R/W-0h						R/W-0h	R/W-0h

Table 4-25. DCDCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	DCDCCTL1	R/W	0h	PMIC_CLKOUT dethering control
1	DCDCLKEN	R/W	0h	PMIC_CLKOUT dethering control block Enable – Multi Bit
0	DCDCRST	R/W	0h	PMIC_CLKOUT dethering control block reset – Multi Bit

4.6.1.12 MISCCTL Register (Offset = 44h) [reset = 0h]

MISCCTL is shown in Figure 4-16 and described in Table 4-26.

Return to Summary Table.

Miscellaneous Control Register

Physical address: 0xFFFF E144

Figure 4-16. MISCCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCTL																															
R/W-0h																															

Table 4-26. MISCCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCTL	R/W	0h	7:0 : Write 0xAD to switch the clock to RCCLK (limp mode) whenever the DCCA gives an error. 10:8 : Write 3'b111 to gate the clock to dbgss. 13:11 : Reserved 14 : rs232_clk_mode SW override value. 23:16 : Write 0xAD to trigger host Interrupt. 31:24 : Write 0xAD to switch the clock to RCCLK (limp mode) whenever the CCCA gives an error.

4.6.1.13 USERMODEEN Register (Offset = 48h) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-17. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-27. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space from 0x00 to 0XFF

4.6.1.14 LVDSPADCTL0 Register (Offset = 4Ch) [reset = 01010101h]

LVDSPADCTL0 is shown in Figure 4-780 and described in Table 4-817.

Return to Summary Table.

LVDS pad control

Figure 4-18. LVDSPADCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL0																															
R/W-01010101h																															

Table 4-28. LVDSPADCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL0	R/W	01010101h	0 : pwrdn Control for i_LVDSclk_io_cell 1 : lopwra Control for i_LVDSclk_io_cell 2 : lopwrb Control for i_LVDSclk_io_cell 3 : lpsel Control for i_LVDSclk_io_cell 4 : sub_lvds_en Control for i_LVDSclk_io_cell 5 : hiz_disable Control for i_LVDSclk_io_cell 6 : ext_res_en Control for i_LVDSclk_io_cell 7 : Reserved 8 : pwrdn Control for i_LVDS_tx0_io_cell 9 : lopwra Control for i_LVDS_tx0_io_cell 10 : lopwrb Control for i_LVDS_tx0_io_cell 11 : lpsel Control for i_LVDS_tx0_io_cell 12 : sub_lvds_en Control for i_LVDS_tx0_io_cell 13 : hiz_disable Control for i_LVDS_tx0_io_cell 14 : ext_res_en Control for i_LVDS_tx0_io_cell 15 : Reserved 16 : pwrdn Control for i_LVDS_tx1_io_cell 17 : lopwra Control for i_LVDS_tx1_io_cell 18 : lopwrb Control for i_LVDS_tx1_io_cell 19 : lpsel Control for i_LVDS_tx1_io_cell 20 : sub_lvds_en Control for i_LVDS_tx1_io_cell 21 : hiz_disable Control for i_LVDS_tx1_io_cell 22 : ext_res_en Control for i_LVDS_tx1_io_cell 23 : Reserved 24 : pwrdn Control for i_LVDS_tx2_io_cell 25 : lopwra Control for i_LVDS_tx2_io_cell 26 : lopwrb Control for i_LVDS_tx2_io_cell 27 : lpsel Control for i_LVDS_tx2_io_cell 28 : sub_lvds_en Control for i_LVDS_tx2_io_cell 29 : hiz_disable Control for i_LVDS_tx2_io_cell 30 : ext_res_en Control for i_LVDS_tx2_io_cell 31 : Reserved

4.6.1.15 LVDSPADCTL1 Register (Offset = 50h) [reset = 101h]

LVDSPADCTL1 is shown in Figure 4-781 and described in Table 4-818.

Return to Summary Table.

LVDS pad control

Figure 4-19. LVDSPADCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL1																															
R/W-101h																															

Table 4-29. LVDSPADCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL1	R/W	101h	0 : pwrdn Control for i_LVDS_tx3_io_cell 1 : lopwra Control for i_LVDS_tx3_io_cell 2 : lopwrb Control for i_LVDS_tx3_io_cell 3 : lpsel Control for i_LVDS_tx3_io_cell 4 : sub_lvds_en Control for i_LVDS_tx3_io_cell 5 : hiz_disable Control for i_LVDS_tx3_io_cell 6 : ext_res_en Control for i_LVDS_tx3_io_cell 7 : Reserved 8 : pwrdn Control for i_LVDSfrclk_io_cell 9 : lopwra Control for i_LVDSfrclk_io_cell 10 : lopwrb Control for i_LVDSfrclk_io_cell 11 : lpsel Control for i_LVDSfrclk_io_cell 12 : sub_lvds_en Control for i_LVDSfrclk_io_cell 13 : hiz_disable Control for i_LVDSfrclk_io_cell 14 : ext_res_en Control for i_LVDSfrclk_io_cell 15 : Reserved 16 : pwrdn Control for i_LVDSvalid_io_cell 17 : lopwra Control fori_LVDSvalid_io_celll 18 : lopwrb Control for i_LVDSvalid_io_cell 19 : lpsel Control for i_LVDSvalid_io_cell 20 : sub_lvds_en Control for i_LVDSvalid_io_cell 21 : hiz_disable Control fori_LVDSvalid_io_cell 22 : ext_res_en Control for i_LVDSvalid_io_cell 23 : Reserved 24 : pwrdn Control for lvds_bias_cell 25 : efuse_set Control for lvds_bias_cell

4.6.1.16 DFTREG0 Register (Offset = 60h) [reset = 0h]

DFTREG0 is shown in Figure 4-782 and described in Table 4-819.

Return to Summary Table.

Figure 4-20. DFTREG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG0																															
R/W-0h																															

Table 4-30. DFTREG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG0	R/W	0h	[3:0] : MSS PBIST SELFTEST KEY = 4'b1010 [4] : Reserved [5] : PBIST IP reset control. 1 = reset, 0 = out of reset. The PBIST_SELFTEST_KEY must be enabled to program this bit. [31:6] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.6.1.17 DFTREG1 Register (Offset = 64h) [reset = 0h]

DFTREG1 is shown in Figure 4-783 and described in Table 4-820.

Return to Summary Table.

Figure 4-21. DFTREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG1																															
R/W-0h																															

Table 4-31. DFTREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG1	R/W	0h	[31:0] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.6.1.18 DFTREG2 Register (Offset = 68h) [reset = 0h]

DFTREG2 is shown in Figure 4-22 and described in Table 4-32.

Return to Summary Table.

Figure 4-22. DFTREG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG2																															
R/W-0h																															

Table 4-32. DFTREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG2	R/W	0h	

4.6.1.19 DFTREG3 Register (Offset = 6Ch) [reset = 0h]

DFTREG3 is shown in Figure 4-23 and described in Table 4-33.

Return to Summary Table.

Figure 4-23. DFTREG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG3																															
R/W-0h																															

Table 4-33. DFTREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG3	R/W	0h	

4.6.1.20 DFTREG4 Register (Offset = 70h) [reset = 0h]

DFTREG4 is shown in Figure 4-24 and described in Table 4-34.

Return to Summary Table.

Figure 4-24. DFTREG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG4																															
R/W-0h																															

Table 4-34. DFTREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG4	R/W	0h	

4.6.1.21 DFTREG5 Register (Offset = 74h) [reset = 0h]

DFTREG5 is shown in Figure 4-784 and described in Table 4-821.

Return to Summary Table.

Figure 4-25. DFTREG5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG5																															
R/W-0h																															

Table 4-35. DFTREG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG5	R/W	0h	

4.6.1.22 SHMEMINITADDR Register (Offset = 78h) [reset = 30000000h]

SHMEMINITADDR is shown in Figure 4-26 and described in Table 4-36.

Return to Summary Table.

Start and End Address for Shared Memory Initialization

Figure 4-26. SHMEMINITADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITENDADDR																INITSTARTADDR															
R/W-3000h																R/W-0h															

Table 4-36. SHMEMINITADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	INITENDADDR	R/W	3000h	End Address for memory initialiation of Shared memory.
15-0	INITSTARTADDR	R/W	0h	Start Address for memory initialization of Shared memory.

4.6.1.23 SHMEMINITECC Register (Offset = 7Ch) [reset = Ch]

SHMEMINITECC is shown in Figure 4-27 and described in Table 4-37.

Return to Summary Table.

ECC value to be written into Shared Memory during initialization.

Figure 4-27. SHMEMINITECC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								INITECCVAL							
								R/W-Ch							

Table 4-37. SHMEMINITECC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU			Reserved
7-0	INITECCVAL	R/W	Ch	ECC value to be written in above start-end address range for shared memory.

4.6.1.24 DSSMEMBANKEN Register (Offset = A0h) [reset = FFFFh]

DSSMEMBANKEN is shown in Figure 4-28 and described in Table 4-38.

Return to Summary Table.

Shared Memory Bank Enable

Figure 4-28. DSSMEMBANKEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																DSSBANKEN															
																R/W-FFFFh															

Table 4-38. DSSMEMBANKEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU			Reserved
15-0	DSSBANKEN	R/W	FFFFh	This bit field defines which all banks are enabled for DSS Set 0th bit to '1' to enable bank0 of shared memory for DSS, 1st bit to enable bank1 and so on.

4.6.1.25 DSSMEMTAB0 Register (Offset = A4h) [reset = 76543210h]

DSSMEMTAB0 is shown in Figure 4-805 and described in Table 4-842.

Return to Summary Table.

Shared Memory Table

Figure 4-29. DSSMEMTAB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSSMEMTAB0																															
R/W-76543210h																															

Table 4-39. DSSMEMTAB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DSSMEMTAB0	R/W	76543210h	DSS memory table for shared memory. Each 4 bit set decides which shared memory bank should be accessed for a given 64K aligned address. For 0th 64KB address, access goes to bank no programmed in [3:0] of this register. For First 64KB address, access goes to bank no programmed in [7:4] of this register. And so on until 7th 64KB address. Value programmed should match with the banks enabled for DSS, decided by register DSSBANKEN.

4.6.1.26 DSSMEMTAB1 Register (Offset = A8h) [reset = FEDCBA98h]

DSSMEMTAB1 is shown in Figure 4-30 and described in Table 4-40.

Return to Summary Table.

Shared Memory Table

Figure 4-30. DSSMEMTAB1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSSMEMTAB1																															
R/W-FEDCBA98h																															

Table 4-40. DSSMEMTAB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DSSMEMTAB1	R/W	FEDCBA98h	Continuation of DSSMEMTAB0. For 8th 64KB to 15th 64KB.

4.6.1.27 TCMAMEMBANK_EN Register (Offset = B0h) [reset = 0h]

TCMAMEMBANK_EN is shown in Figure 4-31 and described in Table 4-41.

Return to Summary Table.

Shared Memory Bank Enable

Figure 4-31. TCMAMEMBANK_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																TCMABANKEN															
R/W-0h																R/W-0h															

Table 4-41. TCMAMEMBANK_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU			Reserved
15-0	TCMABANKEN	R/W	0h	This bit field defines which all banks are enabled for MSS TCMA Set 0th bit to '1' to enable bank0 of shared memory for MSS TCMA, 1st bit to enable bank1 and so forth.

4.6.1.28 TCMAMEMTAB0 Register (Offset = B4h) [reset = 76543210h]

TCMAMEMTAB0 is shown in Figure 4-32 and described in Table 4-42.

Return to Summary Table.

Shared Memory Table

Figure 4-32. TCMAMEMTAB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCMAMEMTAB0																															
R/W-76543210h																															

Table 4-42. TCMAMEMTAB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TCMAMEMTAB0	R/W	76543210h	MSS TCMA memory table for shared memory. Each 4 bit set decides which shared memory bank should be accessed for a given 64K aligned address. For 0th 64KB address, access goes to bank no programmed in [3:0] of this register. For First 64KB address, access goes to bank no programmed in [7:4] of this register. And so on until 7th 64KB address. Value programmed should match with the banks enabled for MSS TCMA decided by register TCMABANKEN.

4.6.1.29 TCMAMEMTAB1 Register (Offset = B8h) [reset = FEDCBA98h]

TCMAMEMTAB1 is shown in Figure 4-33 and described in Table 4-43.

Return to Summary Table.

Shared Memory Table

Figure 4-33. TCMAMEMTAB1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCMAMEMTAB1																															
R/W-FEDCBA98h																															

Table 4-43. TCMAMEMTAB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TCMAMEMTAB1	R/W	FEDCBA98h	Continuation of TCMAMEMTAB0. For 8th 64KB to 15th 64KB.

4.6.1.30 TCMBMEMBANKEN Register (Offset = C0h) [reset = 0h]

TCMBMEMBANKEN is shown in Figure 4-34 and described in Table 4-44.

Return to Summary Table.

Shared Memory Bank Enable

Figure 4-34. TCMBMEMBANKEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																TCMBBANKEN															
																R/W-0h															

Table 4-44. TCMBMEMBANKEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU			Reserved
15-0	TCMBBANKEN	R/W	0h	This bit field defines which all banks are enabled for MSS TCMB Set 0th bit to '1' to enable bank0 of shared memory for MSS TCMB, 1st bit to enable bank1 and so on.

4.6.1.31 TCMBMEMTAB0 Register (Offset = C4h) [reset = 76543210h]

TCMBMEMTAB0 is shown in Figure 4-35 and described in Table 4-45.

Return to Summary Table.

Shared Memory Table

Figure 4-35. TCMBMEMTAB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCMBMEMTAB0																															
R/W-76543210h																															

Table 4-45. TCMBMEMTAB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TCMBMEMTAB0	R/W	76543210h	MSS TCMB memory table for shared memory. Each 4 bit set decides which shared memory bank should be accessed for a given 64K aligned address. For 0th 64KB address, access goes to bank no programmed in [3:0] of this register. For First 64KB address, access goes to bank no programmed in [7:4] of this register. And so on until 7th 64KB address. Value programmed should match with the banks enabled for MSS TCMB decided by register TCMBBANKEN.

4.6.1.32 TCMBMEM_TAB1 Register (Offset = C8h) [reset = FEDCBA98h]

TCMBMEM_TAB1 is shown in Figure 4-36 and described in Table 4-46.

Return to Summary Table.

Shared Memory Table

Figure 4-36. TCMBMEM_TAB1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCMBMEMTAB1																															
R/W-FEDCBA98h																															

Table 4-46. TCMBMEM_TAB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TCMBMEMTAB1	R/W	FEDCBA98h	Continuation of TCMBMEMTAB0. For 8th 64KB to 15th 64KB.

4.6.1.33 MEMINITSTART Register (Offset = D8h) [reset = 0h]

MEMINITSTART is shown in Figure 4-820 and described in Table 4-859.

Return to Summary Table.

Memory Initialization trigger bit for memories in BSS and DSS

Figure 4-37. MEMINITSTART Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							SHMEMSTART
							0h
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							

Table 4-47. MEMINITSTART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	Reserved
16	SHMEMSTART		0h	Writing '1' will trigger Shared memory initialization for the range decided by register SHMEMINITADDR
15-0	RESERVED			Reserved

4.6.1.34 MEMINITDONE Register (Offset = DCh) [reset = 0h]

MEMINITDONE is shown in Figure 4-822 and described in Table 4-861.

Return to Summary Table.

Memory Initialization done status for memories in BSS and DSS

Figure 4-38. MEMINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							SHMEM
							R-0h
15	14	13	12	11	10	9	8
RESERVED						BSSVIMMEM	RESERVED
						R-0h	
7	6	5	4	3	2	1	0
RESERVED							

Table 4-48. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED			Reserved
16	SHMEM	R	0h	Memory Initialization done status for Shared memory for the range decided by register SHMEMINITADDR
15-9	RESERVED			Reserved
9	BSSVIMMEM	R	0h	Memory Initialization done status for BSS VIM memory
8-0	RESERVED	R	0h	Reserved

4.6.1.35 MSS_SIGNATURE Register (Offset = FCh) [reset = 0BB1F02Fh]

MSS_SIGNATURE is shown in Figure 4-787 and described in Table 4-824.

Return to Summary Table.

Spare Register

Figure 4-39. MSS_SIGNATURE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSS_SIGNATURE																															
R/W-0BB1F02Fh																															

Table 4-49. MSS_SIGNATURE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSS_SIGNATURE	R/W	0BB1F02Fh	

4.6.1.36 MISCCTL1 Register (Offset = 178h) [reset = 0h]

MISCCTL1 is shown in Figure 4-789 and described in Table 4-826.

Return to Summary Table.

Miscellaneous Control Register

Figure 4-40. MISCCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCTL1																															
R/W-0h																															

Table 4-50. MISCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCTL1	R/W	0h	7:0 : Write 0xAD to enable Warm_resetn from external to chip also. 16:8 : Write 0xAD to take board level loop back clock for QSPI. 24:16 : Write 0xAD to external clock as QSPI baud clock source – needed for DFT IO char..

4.6.1.37 USERMODEEN2 Register (Offset = 180h) [reset = 0h]

USERMODEEN2 is shown in Figure 4-790 and described in Table 4-827.

Return to Summary Table.

Figure 4-41. USERMODEEN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN2																															
R/W-0h																															

Table 4-51. USERMODEEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN2	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space which are resettable only by Power on reset. i.e. from offset address 0x100 to 0x1FF

4.6.1.38 SYSTICK Register (Offset = 18Ch) [reset = 0h]

SYSTICK is shown in Figure 4-791 and described in Table 4-828.

Return to Summary Table.

Figure 4-42. SYSTICK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSTICK																															
R-0h																															

Table 4-52. SYSTICK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYSTICK	R	0h	Continuous counter running on 32Khz derived from RC clock.

4.6.2 MSS_RCM Registers

Table 4-848 lists the memory-mapped registers for the MSS_RCM. All register offset addresses not listed in Table 4-848 should be considered as reserved locations and the register contents should not be modified.

Table 4-53. MSS_RCM Registers

Offset	Acronym	Register Name	Section
8h	SOFTRST2	SYS_SOFT_RESET2	Section 4.6.2.1
18h	CLKDIVCTL0	CLKDIV	Section 4.6.2.2
1Ch	CLKSRCSEL0	CLKSRCSEL	Section 4.6.2.3
20h	CR4CTL	CR4CTL	Section 4.6.2.4
3Ch	CLKGATE	CLK_GATE	Section 4.6.2.5
44h	CLKSRCSEL1	SYS_CLKSRCSEL	Section 4.6.2.6
54h	CURRCLKDIV0	CURR_CLKDIV	Section 4.6.2.7
5Ch	MEMINITSTART	MEM_INIT_START	Section 4.6.2.8
60h	CURRCLKDIV1	CURR_CLKDIV1	Section 4.6.2.9
6Ch	MEMINITDONE	MEM_INIT_DONE	Section 4.6.2.10
80h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.6.2.11
84h	NSYSPERUSERMODEN	NON_SYS_PERIPH_USERMODEEN	Section 4.6.2.12
90h	ESMGATE0	ESMGATE0	Section 4.6.2.13
94h	ESMGATE1	ESMGATE1	Section 4.6.2.14
98h	ESMGATE2	ESMGATE2	Section 4.6.2.15
9Ch	ESMGATE3	ESMGATE3	Section 4.6.2.16
A0h	ESMGATE4	ESMGATE4	Section 4.6.2.17
ACh	KEY	CFGREG_ACCESS_KEY	Section 4.6.2.18
B8h	SWIRQA	SWIRQ0	Section 4.6.2.19
BCh	SWIRQB	SWIRQ1	Section 4.6.2.20
C0h	MISCCTL0	MISCELLANEOUS_CTL_REG	Section 4.6.2.21
C4h	ATCMERRCAPTCTL	ATCMERRCAPT	Section 4.6.2.22
C8h	B0TCMERRCAPTCTL	B0TCMERRCAPT	Section 4.6.2.23
CCh	B1TCMERRCAPTCTL	B1TCMERRCAPT	Section 4.6.2.24
D0h	SOFTCORERST	SOFT_CORE_RST	Section 4.6.2.25
D8h	RSTCAUSE	MSS_RST_CAUSE	Section 4.6.2.26
DCh	RSTCAUSECLR	MSS_RST_CAUSE_CLR	Section 4.6.2.27
E0h	SPITRIGSRC	SPI_TRIG_SRC	Section 4.6.2.28
E4h	CLKINUSE	MSS_CLK_IN_USE	Section 4.6.2.29
E8h	ECCEN	MSS_ECC_EN	Section 4.6.2.30
ECh	ECCCAPT	MSS_ECC_CAPT	Section 4.6.2.31
F0h	CLKDIVCTL2	CLKDIV2	Section 4.6.2.32
FCh	SWIRQC	SW_IRQ2	Section 4.6.2.33

Complex bit access types are encoded to fit into small table cells. Table 4-849 shows the codes that are used for access types in this section.

Table 4-54. MSS_RCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write

Table 4-54. MSS_RCM Access Type Codes (continued)

Access Type	Code	Description
Reset or Default Value		
-n		Value after reset or the default value

4.6.2.1 SOFTRST2 Register (Offset = 8h) [reset = 0h]

SOFTRST2 is shown in Figure 4-812 and described in Table 4-851.

Return to Summary Table.

Figure 4-43. SOFTRST2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIMRST								RESERVED																							
R/W-0h								R/W-0h																							

Table 4-55. SOFTRST2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VIMRST	R/W	0h	Write 0xAD to assert a VIM only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-0	RESERVED	R/W	0h	Reserved

4.6.2.2 CLKDIVCTL0 Register (Offset = 18h) [reset = 0h]

CLKDIVCTL0 is shown in Figure 4-813 and described in Table 4-852.

Return to Summary Table.

Figure 4-44. CLKDIVCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DCANCLKDIV							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCLKDIV								RESERVED							
R/W-0h								R/W-0h							

Table 4-56. CLKDIVCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R/W	0h	Reserved
23-16	DCANCLKDIV	R/W	0h	Divide value for DCAN source clock selected by field DCANCLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
15-8	VCLKCLKDIV	R/W	0h	Divide value for MSS subsystem source clock selected by field VCLKCLKSRCSEL in register CLKSRCSEL1 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
7-0	RESERVED	R/W	0h	Reserved

4.6.2.3 CLKSRCSEL0 Register (Offset = 1Ch) [reset = 0h]

CLKSRCSEL0 is shown in Figure 4-45 and described in Table 4-57.

Return to Summary Table.

Figure 4-45. CLKSRCSEL0 Register

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				QSPICLKSRCSEL			
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED				FRAYCLKSRCSEL			
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				DCANCLKSRCSEL			
R/W-0h							

Table 4-57. CLKSRCSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED			Reserved
19-16	QSPICLKSRCSEL	R/W	0h	Select clock source for QSPI baud clock 000 => MSS_VCLK 001, 101, 111 => RCCLK 010 => 600-Mhz PLL divided clock 100 => CPUCLK 110 => REFCLK from ANA
15-12	RESERVED			Reserved
11-8	FRAYCLKSRCSEL	R/W	0h	Select clock source for FRAY 000 => MSS_VCLK 001, 101, 111 => RCCLK 010 => 600-Mhz PLL divided clock 100 => CPUCLK 110 => REFCLK from ANA
7-4	RESERVED			Reserved
3-0	DCANCLKSRCSEL	R/W	0h	Select clock source for DCAN 000 => MSS_VCLK 001, 101, 111 => RCCLK 010 => 600-Mhz PLL divided clock 100 => CPUCLK 110 => REFCLK from ANA

4.6.2.4 CR4CTL Register (Offset = 20h) [reset = 0h]

CR4CTL is shown in Figure 4-815 and described in Table 4-854.

Return to Summary Table.

Figure 4-46. CR4CTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								MEMSWAPWAIT							
								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR4MEMSWAP								RESERVED							
R/W-0h															

Table 4-58. CR4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED			Reserved
23-16	MEMSWAPWAIT	R/W	0h	When CR4MEMSWAP is 0xAD : Write 0xAD to this field enable the CR4MEMSWAP only after a CR4 reset – either by writing to CR4SYSRST or by writing to PRCR register in CR4 debug space.
15-8	CR4MEMSWAP	R/W	0h	Write 0xAD will map the MSS CR4 0x0000_0000 to MSS CR4 TCMA RAM start address.
7-0	RESERVED			Reserved.

NOTE: This register used for the ROM eclipsing feature. Refer to the ROM Eclipsing section for more details.

4.6.2.5 CLKGATE Register (Offset = 3Ch) [reset = 400h]

CLKGATE is shown in Figure 4-47 and described in Table 4-59.

Return to Summary Table.

Figure 4-47. CLKGATE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	DCANCLKGATE	QSPICLKGATE	RESERVED		
0h		0h	R/W-0h	R/W-0h	0h		

Table 4-59. CLKGATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED		0h	Reserved
4	DCANCLKGATE	R/W	0h	Pre clock divider gate for DCAN clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
3	QSPICLKGATE	R/W	0h	Pre clock divider gate for QSPI clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
2-0	RESERVED		0h	Reserved

4.6.2.6 CLKSRCSEL1 Register (Offset = 44h) [reset = 0h]

CLKSRCSEL1 is shown in Figure 4-817 and described in Table 4-856.

Return to Summary Table.

Figure 4-48. CLKSRCSEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RESERVED				VCLKCLKSRCSEL			
0h				R/W-0h			

Table 4-60. CLKSRCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED		0h	Reserved
3-0	VCLKCLKSRCSEL	R/W	0h	Select clock source for MSS subsystem VCLK 000 => CPUCLK 001, 100, 101, 111 => RCCLK 010 => 600-Mhz PLL divided clock 100 => CPUCLK 110 => REFCLK from ANA

4.6.2.7 CURRCLKDIV0 Register (Offset = 54h) [reset = 0h]

CURRCLKDIV0 is shown in Figure 4-818 and described in Table 4-857.

Return to Summary Table.

Figure 4-49. CURRCLKDIV0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRAYCURRCLKDIV								DCANCURRCLKDIV							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCURRCLKDIV								RESERVED							
R-0h								R-0h							

Table 4-61. CURRCLKDIV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FRAYCURRCLKDIV	R	0h	Returns Current divide value of FRAY baud clock divider.
23-16	DCANCURRCLKDIV	R	0h	Returns Current divide value of DCAN baud clock divider.
15-8	VCLKCURRCLKDIV	R	0h	Returns Current divide value of VCLK divider.
7-0	RESERVED	R	0h	Reserved

4.6.2.8 MEMINITSTART Register (Offset = 5Ch) [reset = 0h]

MEMINITSTART is shown in Figure 4-820 and described in Table 4-859.

Return to Summary Table.

Figure 4-50. MEMINITSTART Register

31	30	29	28	27	26	25	24
MEMINITKEY							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							BSSMBOX4MSSMEM
							0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	DCANMEM	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-62. MEMINITSTART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MEMINITKEY	R/W	0h	Memory hardware initialization global enable key. Write 0XAD to enable MEMINIT.
23-9	RESERVED			Reserved
8	BSSMBOX4MSSMEM		0h	Writing '1' will trigger BSS mail box fo MSS memory initialization.
7	MSSMBOX4BSSMEM		0h	Writing '1' will trigger MSS mail box fo BSS memory initialization.
6	DCANMEM		0h	Writing '1' will trigger DCAN memory initialization.
5	SPIBMEM		0h	Reserved
4	SPIAMEM		0h	Writing '1' will trigger SPIA memory initialization.
3	VIMMEM		0h	Writing '1' will trigger VIM memory initialization.
2	DMAMEM		0h	Writing '1' will trigger DMA memory initialization.
1	CR4TCMBMEM		0h	Writing '1' will trigger MSS TCMA memory initialization.
0	CR4TCMAMEM		0h	Writing '1' will trigger MSS TCMA memory initialization.

4.6.2.9 CURRCLKDIV1 Register (Offset = 60h) [reset = 0h]

CURRCLKDIV1 is shown in Figure 4-821 and described in Table 4-860.

Return to Summary Table.

Figure 4-51. CURRCLKDIV1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								QSPICURRCLKDIV							
R-0h								R-0h							

Table 4-63. CURRCLKDIV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	QSPICURRCLKDIV	R	0h	Returns Current divide value of QSPI_CLK divider.

4.6.2.10 MEMINITDONE Register (Offset = 6Ch) [reset = 0h]

MEMINITDONE is shown in Figure 4-822 and described in Table 4-861.

Return to Summary Table.

Figure 4-52. MEMINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							BSSMBOX4MSSMEM
							R-0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	DCANMEM	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 4-64. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED			Reserved
8	BSSMBOX4MSSMEM	R	0h	Memory Initialization done status for BSS mailbox f MSS memory
7	MSSMBOX4BSSMEM	R	0h	Memory Initialization done status for MSS mailbox for BSS memory
6	DCANMEM	R	0h	Memory Initialization done status for DCAN memory
5	SPIBMEM	R	0h	Reserved
4	SPIAMEM	R	0h	Memory Initialization done status for MSS SPIA memory
3	VIMMEM	R	0h	Memory Initialization done status for MSS VIM memory
2	DMAMEM	R	0h	Memory Initialization done status for MSS DMA memory
1	CR4TCMBMEM	R	0h	Memory Initialization done status for MSS TCMB memory
0	CR4TCMAMEM	R	0h	Memory Initialization done status for MSS TCMA memory

4.6.2.11 USERMODEEN Register (Offset = 80h) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-53. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-65. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS RCM space.

4.6.2.12 NSYSPERUSERMODEN Register (Offset = 84h) [reset = 0h]

NSYSPERUSERMODEN is shown in Figure 4-828 and described in Table 4-867.

Return to Summary Table.

Figure 4-54. NSYSPERUSERMODEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSYSPERUSERMODEN																															
R/W-0h																															

Table 4-66. NSYSPERUSERMODEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NSYSPERUSERMODEN	R/W	0h	2:0 : Write 3'b111 to enable user mode access to SPIA 5:3 : Write 3'b111 to enable user mode access to SPIB 10:8 : Write 3'b111 to enable user mode access to GIO 13:11 : Write 3'b111 to enable user mode access to QSPI 18:16 : Write 3'b111 to enable user mode access to SCIA 21:19 : Write 3'b111 to enable user mode access to SCIB 26:24 : Write 3'b111 to enable user mode access to DCAN

4.6.2.13 ESMGATE0 Register (Offset = 90h) [reset = 0h]

ESMGATE0 is shown in Figure 4-831 and described in Table 4-870.

Return to Summary Table.

Figure 4-55. ESMGATE0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE0																															
R/W-0h																															

Table 4-67. ESMGATE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE0	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 0 [7:4] : Gate ESM group2 line 1 [11:8] : Gate ESM group2 line 2 [15:12] : Gate ESM group2 line 3 [19:16] : Gate ESM group2 line 4 [23:20] : Gate ESM group2 line 5 [27:24] : Gate ESM group2 line 6 [31:28] : Gate ESM group2 line 7

4.6.2.14 ESMGATE1 Register (Offset = 94h) [reset = 0h]

ESMGATE1 is shown in Figure 4-832 and described in Table 4-871.

Return to Summary Table.

Figure 4-56. ESMGATE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE1																															
R/W-0h																															

Table 4-68. ESMGATE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE1	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 8 [7:4] : Gate ESM group2 line 9 [11:8] : Gate ESM group2 line 10 [15:12] : Gate ESM group2 line 11 [19:16] : Gate ESM group2 line 12 [23:20] : Gate ESM group2 line 13 [27:24] : Gate ESM group2 line 14 [31:28] : Gate ESM group2 line 15

4.6.2.15 ESMGATE2 Register (Offset = 98h) [reset = 0h]

ESMGATE2 is shown in Figure 4-833 and described in Table 4-872.

Return to Summary Table.

Figure 4-57. ESMGATE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE2																															
R/W-0h																															

Table 4-69. ESMGATE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE2	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23

4.6.2.16 ESMGATE3 Register (Offset = 9Ch) [reset = 0h]

ESMGATE3 is shown in Figure 4-834 and described in Table 4-873.

Return to Summary Table.

Figure 4-58. ESMGATE3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE3																															
R/W-0h																															

Table 4-70. ESMGATE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE3	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23

4.6.2.17 ESMGATE4 Register (Offset = A0h) [reset = 0h]

ESMGATE4 is shown in Figure 4-835 and described in Table 4-874.

Return to Summary Table.

Figure 4-59. ESMGATE4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE4																															
R/W-0h																															

Table 4-71. ESMGATE4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE4	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group3 line 0 [7:4] : Gate ESM group3 line 1 [11:8] : Gate ESM group3 line 2 [15:12] : Gate ESM group3 line 3 [19:16] : Gate ESM group3 line 4 [23:20] : Gate ESM group3 line 5 [27:24] : Gate ESM group3 line 6 [31:28] : Gate ESM group3 line 7

4.6.2.18 KEY Register (Offset = ACh) [reset = 83E783E7h]

KEY is shown in Figure 4-836 and described in Table 4-875.

Return to Summary Table.

Figure 4-60. KEY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY																															
R/W-83E783E7h																															

Table 4-72. KEY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY	R/W	83E783E7h	Kicker Register. The value 83E7_83E7h must be written as part of the process to unlock the CPU write access to the MSS RCM registers

4.6.2.19 SWIRQA Register (Offset = B8h) [reset = 0h]

SWIRQA is shown in Figure 4-837 and described in Table 4-876.

Return to Summary Table.

Figure 4-61. SWIRQA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ1								SWIRQ1DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ0								SWIRQ0DAT							
R/W-0h								R/W-0h							

Table 4-73. SWIRQA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ1	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ1DAT	R/W	0h	System software interrupt data. These bits contain user read/write register bits. They may be used by the application software as different entry points for the interrupt routine.
15-8	SWIRQ0	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ0DAT	R/W	0h	System software interrupt data. These bits contain user read/write register bits. They may be used by the application software as different entry points for the interrupt routine.

4.6.2.20 SWIRQB Register (Offset = BCh) [reset = 0h]

SWIRQB is shown in Figure 4-838 and described in Table 4-877.

Return to Summary Table.

Figure 4-62. SWIRQB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ3								SWIRQ3DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ2								SWIRQ2DAT							
R/W-0h								R/W-0h							

Table 4-74. SWIRQB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ3	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ3DAT	R/W	0h	System software interrupt data. These bits contain user read/write register bits. They may be used by the application software as different entry points for the interrupt routine.
15-8	SWIRQ2	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ2DAT	R/W	0h	System software interrupt data. These bits contain user read/write register bits. They may be used by the application software as different entry points for the interrupt routine.

4.6.2.21 MISCCTL0 Register (Offset = C0h) [reset = ADADh]

MISCCTL0 is shown in Figure 4-839 and described in Table 4-878.

Return to Summary Table.

Figure 4-63. MISCCTL0 Register

31	30	29	28	27	26	25	24
RESERVED							TCMB1EZDIS
							R/W-0h
23	22	21	20	19	18	17	16
TCMB1EZDIS			TCMB0EZDIS			TCMAEZDIS	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED							

Table 4-75. MISCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED			Reserved
24-22	TCMB1EZDIS	R/W	0h	Write 111 to make TCMB1 EZ to '1' regardless of Functional value.
21-19	TCMB0EZDIS	R/W	0h	Write 111 to make TCMB0 EZ to '1' regardless of Functional value.
18-16	TCMAEZDIS	R/W	0h	Write 111 to make TCMA EZ to '1' regardless of Functional value.
15-0	RESERVED			Reserved

4.6.2.22 ATCMERRCAPTCTL Register (Offset = C4h) [reset = 0h]

ATCMERRCAPTCTL is shown in Figure 4-840 and described in Table 4-879.

Return to Summary Table.

Figure 4-64. ATCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRATCADDR			
				R-0h			
23	22	21	20	19	18	17	16
ERRATCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRATCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		ATCFORCEERR			ERRATCADDRCLR		
R/W-0h					0h		

Table 4-76. ATCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1			Reserved
27-8	ERRATCADDR	R	0h	TCM address for which parity error happened
7-6	NU0			Reserved
5-3	ATCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRATCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to enable the latching.

4.6.2.23 B0TCMERRCAPTCTL Register (Offset = C8h) [reset = 0h]

B0TCMERRCAPTCTL is shown in Figure 4-841 and described in Table 4-880.

Return to Summary Table.

Figure 4-65. B0TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB0TCADDR			
				R-0h			
23	22	21	20	19	18	17	16
ERRB0TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB0TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B0TCFORCEERR			ERRB0TCADDRCLR		
R/W-0h					0h		

Table 4-77. B0TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1			Reserved
27-8	ERRB0TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0			Reserved
5-3	B0TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB0TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to enable the latching.

4.6.2.24 B1TCMERRCAPTCTL Register (Offset = CCh) [reset = 0h]

B1TCMERRCAPTCTL is shown in Figure 4-842 and described in Table 4-881.

Return to Summary Table.

Figure 4-66. B1TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB1TCADDR			
				R-0h			
23	22	21	20	19	18	17	16
ERRB1TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB1TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B1TCFORCEERR			ERRB1TCADDRCLR		
R/W-0h					0h		

Table 4-78. B1TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1			Reserved
27-8	ERRB1TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0			Reserved
5-3	B1TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB1TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to enable the latching.

4.6.2.25 SOFTCORERST Register (Offset = D0h) [reset = 000F0F00h]

SOFTCORERST is shown in Figure 4-843 and described in Table 4-882.

Return to Summary Table.

Figure 4-67. SOFTCORERST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RST_WFICHECKEN								RESERVED							
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSTTOASSRTDLY								RESERVED							
R/W-0Fh															

Table 4-79. SOFTCORERST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RST_WFICHECKEN	R/W	0h	0xAD : When CR4SYSRST is set, Before asserting the reset to CR4 wait for CR4 to enter WFI. 0X0: Dont wait for WFI
23-16	RESERVED			Reserved
15-8	RSTTOASSRTDLY	R/W	0Fh	Wait for programmed number of clock cycle before reset is asserted to CR4.
7-0	RESERVED			Reserved

4.6.2.26 RSTCAUSE Register (Offset = D8h) [reset = 0h]

RSTCAUSE is shown in Figure 4-844 and described in Table 4-883.

Return to Summary Table.

Figure 4-68. RSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								RSTCAUSE							
R-0h																															

Table 4-80. RSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU			Reserved
7-0	RSTCAUSE	R	0h	MSS_RCM: RSTCAUSE 0000_1001 : System out of NRESET 0000_1000 : System out of Warm Reset 0010_0000 : CR4 reset because of Software trigger. 0001_0000 : CR4 STC reset 0100_0000 : CR4 reset because of writing to PRCR register in CR4 debug space. Note: Because the ROM Bootloader issues a soft reset to pass control to the application (or secondary bootloader), this register always reads 0x20. The reset value is stored by the ROM Bootloader into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.6.2.27 RSTCAUSECLR Register (Offset = DCh) [reset = 0h]

RSTCAUSECLR is shown in Figure 4-845 and described in Table 4-884.

Return to Summary Table.

Figure 4-69. RSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								RSTCAUSECLR							
								0h							

Table 4-81. RSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU			
7-0	RSTCAUSECLR		0h	Write 0xAD to clear RSTCAUSE.

4.6.2.28 SPITRIGSRC Register (Offset = E0h) [reset = 0h]

SPITRIGSRC is shown in Figure 4-846 and described in Table 4-885.

Return to Summary Table.

Figure 4-70. SPITRIGSRC Register

31	30	29	28	27	26	25	24
NU4							SPIBTRIG1
							R/W-0h
23	22	21	20	19	18	17	16
NU3							SPIBTRIG0
							R/W-0h
15	14	13	12	11	10	9	8
NU2							SPIATRIG1
							R/W-0h
7	6	5	4	3	2	1	0
NU1							SPIATRIG0
							R/W-0h

Table 4-82. SPITRIGSRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4			
24	SPIBTRIG1	R/W	0h	Reserved
23-17	NU3			
16	SPIBTRIG0	R/W	0h	Reserved
15-9	NU2			
8	SPIATRIG1	R/W	0h	1st bit ofTRG_SRC input of SPIA.
7-1	NU1			
0	SPIATRIG0	R/W	0h	0th bit ofTRG_SRC input of SPIA.

4.6.2.29 CLKINUSE Register (Offset = E4h) [reset = 0h]

CLKINUSE is shown in Figure 4-71 and described in Table 4-83.

Return to Summary Table.

Figure 4-71. CLKINUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QSPICLKINUSE				DCANCLKINUSE				FRAYCLKINUSE				VCLKINUSE			
R-0h				R-0h				R-0h				R-0h			

Table 4-83. CLKINUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED			Reserved
15-12	QSPICLKINUSE	R	0h	Current Clock Source Select Mux value for QSPI CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => XTAL clock 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
11-8	DCANCLKINUSE	R	0h	Current Clock Source Select Mux value for DCAN CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => XTAL clock 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
7-4	FRAYCLKINUSE	R	0h	Current Clock Source Select Mux value for FRAY CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => XTAL clock 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz) Not valid for 14XX. No FRAY
3-0	VCLKINUSE	R	0h	Current Clock Source Select Mux value for VCLK 000 => XTAL clock (40Mhz or 50 Mhz or 80Mh or 100Mh) 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 101 => RCCLK (10Mhz) 110 => REFCLK 111 => RCCLK (10Mhz)

4.6.2.30 ECCEN Register (Offset = E8h) [reset = 0h]

ECCEN is shown in Figure 4-72 and described in Table 4-84.

Return to Summary Table.

Figure 4-72. ECCEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCEN																															
R/W-0h																															

Table 4-84. ECCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCEN	R/W	0h	7:0 : Writing 0xAD will enable ECC for MSS mailbox for BSS 15:8: Writing 0xAD will enable ECC for BSS mailbox for MSS 18:16 : Write 3'b111 to clear the Address captured because of ECC error in MSS mailbox for MSS 21:19: Write 3'b111 to clear the Address captured because of ECC error. In BSS mailbox for MSS

4.6.2.31 ECCCAPT Register (Offset = ECh) [reset = 0h]

ECCCAPT is shown in Figure 4-73 and described in Table 4-85.

Return to Summary Table.

Figure 4-73. ECCCAPT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPT																															
R-0h																															

Table 4-85. ECCCAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPT	R	0h	7:0 : mss_mbox4bss_ecc_fault_address 14:8 : mss_mbox4bss_repaired_bit 23:16 : mss_mbox4bss_ecc_fault_address 30:24 : mss_mbox4bss_repaired_bit

4.6.2.32 CLKDIVCTL2 Register (Offset = F0h) [reset = 0h]

CLKDIVCTL2 is shown in Figure 4-850 and described in Table 4-889.

Return to Summary Table.

Figure 4-74. CLKDIVCTL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								QSPICLKDIV							
								R/W-0h							

Table 4-86. CLKDIVCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU			
7-0	QSPICLKDIV	R/W	0h	Divide value for QSPI baud clock selected by field QSPICLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div255

4.6.2.33 SWIRQC Register (Offset = FCh) [reset = 0h]

SWIRQC is shown in Figure 4-851 and described in Table 4-890.

Return to Summary Table.

Figure 4-75. SWIRQC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ5								SWIRQ5DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ4								SWIRQ4DAT							
R/W-0h								R/W-0h							

Table 4-87. SWIRQC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ5	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ5DAT	R/W	0h	System software interrupt data. These bits contain user read/write register bits. They may be used by the application software as different entry points for the interrupt routine.
15-8	SWIRQ4	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ4DAT	R/W	0h	System software interrupt data. These bits contain user read/write register bits. They may be used by the application software as different entry points for the interrupt routine.

4.6.3 MSS_GPCFG_REG Registers

Table 4-891 lists the memory-mapped registers for the MSS_GPCFG_REG. All register offset addresses not listed in Table 4-891 should be considered as reserved locations and the register contents should not be modified.

Table 4-88. MSS_GPCFG_REG Registers

Offset	Acronym	Register Name	Section
0h	GPCFG0	GPCFG0	Section 4.6.3.1
4h	GPCFG1	GPCFG1	Section 4.6.3.2
8h	GPCFG2	GPCFG2	Section 4.6.3.3
Ch	GPCFG3	GPCFG3	Section 4.6.3.4
10h	GPCFG4	GPCFG4	Section 4.6.3.5
D0h	CCCACFG0	CCCA_CFG0	Section 4.6.3.6
D4h	CCCACFG1	CCCA_CFG1	Section 4.6.3.7
D8h	CCCACFG2	CCCA_CFG2	Section 4.6.3.8
DCh	CCCACFG3	CCCA_CFG3	Section 4.6.3.9
E0h	CCCBCFG0	CCCB_CFG0	Section 4.6.3.10
E4h	CCCBCFG1	CCCB_CFG1	Section 4.6.3.11
E8h	CCCBCFG2	CCCB_CFG2	Section 4.6.3.12
ECh	CCCBCFG3	CCCB_CFG3	Section 4.6.3.13
F0h	CCCACNTVAL	CCCACNTVAL	Section 4.6.3.14
F4h	CCCBCNTVAL	CCCBCNTVAL	Section 4.6.3.15
F8h	CCCABERRSTAT	CCCABERRSTAT	Section 4.6.3.16
FCh	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.6.3.17

Complex bit access types are encoded to fit into small table cells. Table 4-892 shows the codes that are used for access types in this section.

Table 4-89. MSS_GPCFG_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.6.3.1 GPCFG0 Register (Offset = 0h) [reset = 0h]

GPCFG0 is shown in Figure 4-852 and described in Table 4-893.

Return to Summary Table.

Figure 4-76. GPCFG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG0																															
R/W-0h																															

Table 4-90. GPCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG0	R/W	0h	General Purpose config register for SW use.

4.6.3.2 GPCFG1 Register (Offset = 4h) [reset = 0h]

GPCFG1 is shown in Figure 4-853 and described in Table 4-894.

Return to Summary Table.

Figure 4-77. GPCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG1																															
R/W-0h																															

Table 4-91. GPCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG1	R/W	0h	General Purpose config register for SW use.

4.6.3.3 GPCFG2 Register (Offset = 8h) [reset = 0h]

GPCFG2 is shown in Figure 4-854 and described in Table 4-895.

Return to Summary Table.

Figure 4-78. GPCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG2																															
R/W-0h																															

Table 4-92. GPCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG2	R/W	0h	General Purpose config register for SW use.

4.6.3.4 GPCFG3 Register (Offset = Ch) [reset = 0h]

GPCFG3 is shown in Figure 4-855 and described in Table 4-896.

Return to Summary Table.

Figure 4-79. GPCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG3																															
R/W-0h																															

Table 4-93. GPCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG3	R/W	0h	General Purpose config register for SW use.

4.6.3.5 GPCFG4 Register (Offset = 10h) [reset = 0h]

GPCFG4 is shown in Figure 4-856 and described in Table 4-897.

Return to Summary Table.

Figure 4-80. GPCFG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG4																															
R/W-0h																															

Table 4-94. GPCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG4	R/W	0h	General Purpose config register for SW use.

4.6.3.6 CCCACFG0 Register (Offset = D0h) [reset = 0h]

CCCACFG0 is shown in Figure 4-859 and described in Table 4-900.

Return to Summary Table.

Figure 4-81. CCCACFG0 Register

31	30	29	28	27	26	25	24
MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU							SINGLE_SHOT_MODE
							R/W-0h
7	6	5	4	3	2	1	0
ENABLE_MODULE	DISABLE_CLOCKS	CLOCK1_SEL			CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-95. CCCACFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU			
8	SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	ENABLE_MODULE	R/W	0h	Enable for CCC
6	DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.6.3.7 CCCACFG1 Register (Offset = D4h) [reset = 0h]

CCCACFG1 is shown in Figure 4-860 and described in Table 4-901.

Return to Summary Table.

Figure 4-82. CCCACFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG1																															
R/W-0h																															

Table 4-96. CCCACFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG1	R/W	0h	count0_expiry_val

4.6.3.8 CCCACFG2 Register (Offset = D8h) [reset = 0h]

CCCACFG2 is shown in Figure 4-861 and described in Table 4-902.

Return to Summary Table.

Figure 4-83. CCCACFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG2																															
R/W-0h																															

Table 4-97. CCCACFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG2	R/W	0h	count1_expected_val

4.6.3.9 CCCACFG3 Register (Offset = DCh) [reset = 0h]

CCCACFG3 is shown in Figure 4-862 and described in Table 4-903.

Return to Summary Table.

Figure 4-84. CCCACFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG3																															
R/W-0h																															

Table 4-98. CCCACFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG3	R/W	0h	Error Counter value in counter1 clock

4.6.3.10 CCCBCFG0 Register (Offset = E0h) [reset = 0h]

CCCBCFG0 is shown in Figure 4-863 and described in Table 4-904.

Return to Summary Table.

Figure 4-85. CCCBCFG0 Register

31	30	29	28	27	26	25	24
MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU							SINGLE_SHOT_MODE
							R/W-0h
7	6	5	4	3	2	1	0
ENABLE_MODULE	DISABLE_CLOCKS	CLOCK1_SEL			CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-99. CCCBCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU			
8	SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	ENABLE_MODULE	R/W	0h	Enable for CCC
6	DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.6.3.11 CCCBCFG1 Register (Offset = E4h) [reset = 0h]

CCCBCFG1 is shown in Figure 4-864 and described in Table 4-905.

Return to Summary Table.

Figure 4-86. CCCBCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG1																															
R/W-0h																															

Table 4-100. CCCBCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG1	R/W	0h	count0_expiry_val

4.6.3.12 CCCBCFG2 Register (Offset = E8h) [reset = 0h]

CCCBCFG2 is shown in Figure 4-865 and described in Table 4-906.

Return to Summary Table.

Figure 4-87. CCCBCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG2																															
R/W-0h																															

Table 4-101. CCCBCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG2	R/W	0h	count1_expected_val

4.6.3.13 CCCBCFG3 Register (Offset = ECh) [reset = 0h]

CCCBCFG3 is shown in Figure 4-866 and described in Table 4-907.

Return to Summary Table.

Figure 4-88. CCCBCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG3																															
R/W-0h																															

Table 4-102. CCCBCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG3	R/W	0h	Error Counter value in counter1 clock

4.6.3.14 CCCACNTVAL Register (Offset = F0h) [reset = 0h]

CCCACNTVAL is shown in Figure 4-867 and described in Table 4-908.

Return to Summary Table.

Figure 4-89. CCCACNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACNTVAL																															
R-0h																															

Table 4-103. CCCACNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACNTVAL	R	0h	count1_val_out

4.6.3.15 CCCBCNTVAL Register (Offset = F4h) [reset = 0h]

CCCBCNTVAL is shown in Figure 4-868 and described in Table 4-909.

Return to Summary Table.

Figure 4-90. CCCBCNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCNTVAL																															
R-0h																															

Table 4-104. CCCBCNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCNTVAL	R	0h	count1_val_out

4.6.3.16 CCCABERRSTAT Register (Offset = F8h) [reset = 0h]

CCCABERRSTAT is shown in Figure 4-869 and described in Table 4-910.

Return to Summary Table.

Figure 4-91. CCCABERRSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCABERRSTAT																															
R-0h																															

Table 4-105. CCCABERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCABERRSTAT	R	0h	7:0 : CCCA Error Status 15:8 : CCCB Error Status

4.6.3.17 USERMODEEN Register (Offset = FCh) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-92. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-106. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS GPCFG space.

4.6.4 DSS_REG Registers

Table 4-927 lists the memory-mapped registers for the DSS_REG. All register offset addresses not listed in Table 4-927 should be considered as reserved locations and the register contents should not be modified.

Table 4-107. DSS_REG Registers

Offset	Acronym	Register Name	Section
50h	RTIEVENTCAPTURESEL	RTIEVENTCAPTURESEL	Section 4.6.4.1
5Ch	ADCBUFCFG1	ADCBUFCFG1	Section 4.6.4.2
60h	ADCBUFCFG2	ADCBUFCFG2	Section 4.6.4.3
64h	ADCBUFCFG3	ADCBUFCFG3	Section 4.6.4.4
68h	ADCBUFCFG4	ADCBUFCFG4	Section 4.6.4.5
6Ch	CQCFG1	CQCFG1	Section 4.6.4.6
80h	TPCCPARSTATCFG	TPCCPARSTATCFG	Section 4.6.4.7
84h	CSI2TXPARSTATCFG	CSI2TXPARSTATCFG	Section 4.6.4.8
A0h	CSICFG1	CSICFG1	Section 4.6.4.9
104h	TPTC0WRMPUSTADD0	TPTC0WRMPUSTADD0	Section 4.6.4.10
108h	TPTC0WRMPUSTADD1	TPTC0WRMPUSTADD1	Section 4.6.4.11
10Ch	TPTC0WRMPUSTADD2	TPTC0WRMPUSTADD2	Section 4.6.4.12
110h	TPTC0WRMPUSTADD3	TPTC0WRMPUSTADD3	Section 4.6.4.13
114h	TPTC0WRMPUSTADD4	TPTC0WRMPUSTADD4	Section 4.6.4.14
118h	TPTC0WRMPUSTADD5	TPTC0WRMPUSTADD5	Section 4.6.4.15
11Ch	TPTC0WRMPUSTADD6	TPTC0WRMPUSTADD6	Section 4.6.4.16
120h	TPTC0WRMPUSTADD7	TPTC0WRMPUSTADD7	Section 4.6.4.17
124h	TPTC0WRMPUENDADD0	TPTC0WRMPUENDADD0	Section 4.6.4.18
128h	TPTC0WRMPUENDADD1	TPTC0WRMPUENDADD1	Section 4.6.4.19
12Ch	TPTC0WRMPUENDADD2	TPTC0WRMPUENDADD2	Section 4.6.4.20
130h	TPTC0WRMPUENDADD3	TPTC0WRMPUENDADD3	Section 4.6.4.21
134h	TPTC0WRMPUENDADD4	TPTC0WRMPUENDADD4	Section 4.6.4.22
138h	TPTC0WRMPUENDADD5	TPTC0WRMPUENDADD5	Section 4.6.4.23
13Ch	TPTC0WRMPUENDADD6	TPTC0WRMPUENDADD6	Section 4.6.4.24
140h	TPTC0WRMPUENDADD7	TPTC0WRMPUENDADD7	Section 4.6.4.25
144h	TPTC0WRMPUERRADD	TPTC0WRMPUERRADD	Section 4.6.4.26
148h	TPTC0RDMPUSTADD0	TPTC0RDMPUSTADD0	Section 4.6.4.27
14Ch	TPTC0RDMPUSTADD1	TPTC0RDMPUSTADD1	Section 4.6.4.28
150h	TPTC0RDMPUSTADD2	TPTC0RDMPUSTADD2	Section 4.6.4.29
154h	TPTC0RDMPUSTADD3	TPTC0RDMPUSTADD3	Section 4.6.4.30
158h	TPTC0RDMPUSTADD4	TPTC0RDMPUSTADD4	Section 4.6.4.31
15Ch	TPTC0RDMPUSTADD5	TPTC0RDMPUSTADD5	Section 4.6.4.32
160h	TPTC0RDMPUSTADD6	TPTC0RDMPUSTADD6	Section 4.6.4.33
164h	TPTC0RDMPUSTADD7	TPTC0RDMPUSTADD7	Section 4.6.4.34
168h	TPTC0RDMPUENDADD0	TPTC0RDMPUENDADD0	Section 4.6.4.35
16Ch	TPTC0RDMPUENDADD1	TPTC0RDMPUENDADD1	Section 4.6.4.36
170h	TPTC0RDMPUENDADD2	TPTC0RDMPUENDADD2	Section 4.6.4.37
174h	TPTC0RDMPUENDADD3	TPTC0RDMPUENDADD3	Section 4.6.4.38
178h	TPTC0RDMPUENDADD4	TPTC0RDMPUENDADD4	Section 4.6.4.39
17Ch	TPTC0RDMPUENDADD5	TPTC0RDMPUENDADD5	Section 4.6.4.40
180h	TPTC0RDMPUENDADD6	TPTC0RDMPUENDADD6	Section 4.6.4.41
184h	TPTC0RDMPUENDADD7	TPTC0RDMPUENDADD7	Section 4.6.4.42
188h	TPTC0RDMPUERRADD	TPTC0RDMPUERRADD	Section 4.6.4.43

Table 4-107. DSS_REG Registers (continued)

Offset	Acronym	Register Name	Section
18Ch	TPTC1WRMPUSTADD0	TPTC1WRMPUSTADD0	Section 4.6.4.44
190h	TPTC1WRMPUSTADD1	TPTC1WRMPUSTADD1	Section 4.6.4.45
194h	TPTC1WRMPUSTADD2	TPTC1WRMPUSTADD2	Section 4.6.4.46
198h	TPTC1WRMPUSTADD3	TPTC1WRMPUSTADD3	Section 4.6.4.47
19Ch	TPTC1WRMPUSTADD4	TPTC1WRMPUSTADD4	Section 4.6.4.48
1A0h	TPTC1WRMPUSTADD5	TPTC1WRMPUSTADD5	Section 4.6.4.49
1A4h	TPTC1WRMPUSTADD6	TPTC1WRMPUSTADD6	Section 4.6.4.50
1A8h	TPTC1WRMPUSTADD7	TPTC1WRMPUSTADD7	Section 4.6.4.51
1ACh	TPTC1WRMPUENDADD0	TPTC1WRMPUENDADD0	Section 4.6.4.52
1B0h	TPTC1WRMPUENDADD1	TPTC1WRMPUENDADD1	Section 4.6.4.53
1B4h	TPTC1WRMPUENDADD2	TPTC1WRMPUENDADD2	Section 4.6.4.54
1B8h	TPTC1WRMPUENDADD3	TPTC1WRMPUENDADD3	Section 4.6.4.55
1BCh	TPTC1WRMPUENDADD4	TPTC1WRMPUENDADD4	Section 4.6.4.56
1C0h	TPTC1WRMPUENDADD5	TPTC1WRMPUENDADD5	Section 4.6.4.57
1C4h	TPTC1WRMPUENDADD6	TPTC1WRMPUENDADD6	Section 4.6.4.58
1C8h	TPTC1WRMPUENDADD7	TPTC1WRMPUENDADD7	Section 4.6.4.59
1CCh	TPTC1WRMPUERRADD	TPTC1WRMPUERRADD	Section 4.6.4.60
1D0h	TPTC1RDMPUSTADD0	TPTC1RDMPUSTADD0	Section 4.6.4.61
1D4h	TPTC1RDMPUSTADD1	TPTC1RDMPUSTADD1	Section 4.6.4.62
1D8h	TPTC1RDMPUSTADD2	TPTC1RDMPUSTADD2	Section 4.6.4.63
1DCh	TPTC1RDMPUSTADD3	TPTC1RDMPUSTADD3	Section 4.6.4.64
1E0h	TPTC1RDMPUSTADD4	TPTC1RDMPUSTADD4	Section 4.6.4.65
1E4h	TPTC1RDMPUSTADD5	TPTC1RDMPUSTADD5	Section 4.6.4.66
1E8h	TPTC1RDMPUSTADD6	TPTC1RDMPUSTADD6	Section 4.6.4.67
1ECh	TPTC1RDMPUSTADD7	TPTC1RDMPUSTADD7	Section 4.6.4.68
1F0h	TPTC1RDMPUENDADD0	TPTC1RDMPUENDADD0	Section 4.6.4.69
1F4h	TPTC1RDMPUENDADD1	TPTC1RDMPUENDADD1	Section 4.6.4.70
1F8h	TPTC1RDMPUENDADD2	TPTC1RDMPUENDADD2	Section 4.6.4.71
1FCh	TPTC1RDMPUENDADD3	TPTC1RDMPUENDADD3	Section 4.6.4.72
200h	TPTC1RDMPUENDADD4	TPTC1RDMPUENDADD4	Section 4.6.4.73
204h	TPTC1RDMPUENDADD5	TPTC1RDMPUENDADD5	Section 4.6.4.74
208h	TPTC1RDMPUENDADD6	TPTC1RDMPUENDADD6	Section 4.6.4.75
20Ch	TPTC1RDMPUENDADD7	TPTC1RDMPUENDADD7	Section 4.6.4.76
210h	TPTC1RDMPUERRADD	TPTC1RDMPUERRADD	Section 4.6.4.77
214h	TPTCMPUVALIDCFG	TPTCMPUVALIDCFG	Section 4.6.4.78
218h	TPTCMPUENCFG	TPTCMPUENCFG	Section 4.6.4.79
21Ch	TESTPATTERNRX1ICFG	TESTPATTERNRX1ICFG	Section 4.6.4.80
220h	TESTPATTERNRX2ICFG	TESTPATTERNRX2ICFG	Section 4.6.4.81
224h	TESTPATTERNRX3ICFG	TESTPATTERNRX3ICFG	Section 4.6.4.82
228h	TESTPATTERNRX4ICFG	TESTPATTERNRX4ICFG	Section 4.6.4.83
22Ch	TESTPATTERNRX1QCFG	TESTPATTERNRX1QCFG	Section 4.6.4.84
230h	TESTPATTERNRX2QCFG	TESTPATTERNRX2QCFG	Section 4.6.4.85
234h	TESTPATTERNRX3QCFG	TESTPATTERNRX3QCFG	Section 4.6.4.86
238h	TESTPATTERNRX4QCFG	TESTPATTERNRX4QCFG	Section 4.6.4.87
23Ch	TESTPATTERNVLDCFG	TESTPATTERNVLDCFG	Section 4.6.4.88
240h	DSSMISC	DSSMISC	Section 4.6.4.89
244h	DSSMISC2	DSSMISC2	Section 4.6.4.90

Complex bit access types are encoded to fit into small table cells. Table 4-928 shows the codes that are used for access types in this section.

Table 4-108. DSS_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.6.4.1 RTIEVENTCAPTURESEL Register (Offset = 50h) [reset = 0h]

RTIEVENTCAPTURESEL is shown in Figure 4-886 and described in Table 4-929.

Return to Summary Table.

Figure 4-93. RTIEVENTCAPTURESEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2												EVT1				NU1												EVT0			
R-												R/W-0h				R-												R/W-0h			

Table 4-109. RTIEVENTCAPTURESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU2	R		
19-16	EVT1	R/W	0h	Not used in 14xx
15-4	NU1	R		
3-0	EVT0	R/W	0h	Not used in 14xx

4.6.4.2 ADCBUFCFG1 Register (Offset = 5Ch) [reset = 00010000h]

ADCBUFCFG1 is shown in Figure 4-988 and described in Table 4-1031.

Return to Summary Table.

Figure 4-94. ADCBUFCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED		ADCBUFNUMCHRP					RESERVED
R/W-0h		R/W-0h					R-1h
15	14	13	12	11	10	9	8
ADCBUFCONTSTOPPL	ADCBUFCONTSTRTPL	ADCBUFCONTMODEEN	ADCBUFWRITEMODE	RESERVED		RX3EN	RX2EN
0h	0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RX1EN	RX0EN	ADCBUFIQSWAP	RESERVED		ADCBUFREALONLYMODE	RESERVED	
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 4-110. ADCBUFCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	Reserved
21-17	ADCBUFNUMCHRP	R/W	0h	Number of chirps to be stored in each Ping and Pong buffer. This register should be programmed with one less than the actual number needed.
16	RESERVED	R	1h	Reserved
15	ADCBUFCONTSTOPPL		0h	Stop Pulse for Continuous mode. The data capture will stop once this register is set.
14	ADCBUFCONTSTRTPL		0h	Start Pulse for Continuous mode. The data capture will start from Address 0 once this register is set. All the other configurations like Enable, Sample Count are expected to be programmed before this pulse.
13	ADCBUFCONTMODEEN	R/W	0h	Continuous mode enable for ADC Buffer. This is set when a fixed number of samples have to be stored in Ping/Pong and not depend on Chirp time-lines (Eg: Analog Lab characterization to stream out continuous data from DFE).
12	ADCBUFWRITEMODE	R/W	0h	0 --> Interleaved, 1 --> Non-interleaved
11-10	RESERVED	R/W	0h	Reserved
9	RX3EN	R/W	0h	Enable for Rx3 write
8	RX2EN	R/W	0h	Enable for Rx2 write
7	RX1EN	R/W	0h	Enable for Rx1 write
6	RX0EN	R/W	0h	Enable for Rx0 write
5	ADCBUFIQSWAP	R/W	0h	0 -->[I is stored in LSB part and Q is stored in MSB part]. 1 --> [Q is stored in LSB and I in MSB]
4-3	RESERVED	R/W	0h	Reserved
2	ADCBUFREALONLYMODE	R/W	0h	0-->Complex Data mode, 1-->Real data mode
1-0	RESERVED	R/W	0h	Reserved

4.6.4.3 ADCBUFCFG2 Register (Offset = 60h) [reset = 01000000h]

ADCBUFCFG2 is shown in Figure 4-989 and described in Table 4-1032.

Return to Summary Table.

Figure 4-95. ADCBUFCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2						ADCBUFADDRX1									
R-						R/W-100h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1						ADCBUFADDRX0									
R-						R/W-0h									

Table 4-111. ADCBUFCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU2	R		
25-16	ADCBUFADDRX1	R/W	100h	Address offset to be added to the internal address pointer for Rx1 writes in Non-interleaved mode.
15-10	NU1	R		
9-0	ADCBUFADDRX0	R/W	0h	Address offset to be added to the internal address pointer for Rx0 writes in Non-interleaved mode.

4.6.4.4 ADCBUFCFG3 Register (Offset = 64h) [reset = 03000200h]

ADCBUFCFG3 is shown in Figure 4-990 and described in Table 4-1033.

Return to Summary Table.

Figure 4-96. ADCBUFCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2						ADCBUFADDRX3									
R-						R/W-300h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1						ADCBUFADDRX2									
R-						R/W-200h									

Table 4-112. ADCBUFCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU2	R		
25-16	ADCBUFADDRX3	R/W	300h	Address offset to be added to the internal address pointer for Rx3 writes in Non-interleaved mode.
15-10	NU1	R		
9-0	ADCBUFADDRX2	R/W	200h	Address offset to be added to the internal address pointer for Rx2 writes in Non-interleaved mode.

4.6.4.5 ADCBUFCFG4 Register (Offset = 68h) [reset = 400h]

ADCBUFCFG4 is shown in Figure 4-991 and described in Table 4-1034.

Return to Summary Table.

Figure 4-97. ADCBUFCFG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																		ADCBUFSAMPCNT													
R-																		R/W-400h													

Table 4-113. ADCBUFCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	NU	R		
13-0	ADCBUFSAMPCNT	R/W	400h	No of samples to store in each Ping and Pong register in continuous mode of ADC Buffer. In real only mode this refers to the number of real samples and in complex mode, this refers to number of complex samples. This refers to the number of samples per channel. This counter increments once for every new sample from DFE (as long as 1 or more channels are enabled). The max allowed value varies depending on other configurations (No of channels enabled and real/complex data). Ex: It is 8192 for real only mode one channel, and 4096 for complex mode one channel. It is only 2048 for real mode 4 channels and 1024 for complex mode 4 channels.

4.6.4.6 CQCFG1 Register (Offset = 6Ch) [reset = 40200000h]

CQCFG1 is shown in Figure 4-887 and described in Table 4-930.

Return to Summary Table.

Figure 4-98. CQCFG1 Register

31	30	29	28	27	26	25	24
NU3	CQ2BASEADDR						
R-	R/W-40h						
23	22	21	20	19	18	17	16
NU2	CQ1BASEADDR						
R-	R/W-20h						
15	14	13	12	11	10	9	8
NU1					CQ0BASEADDR		
R-					R/W-0h		
7	6	5	4	3	2	1	0
CQ0BASEADDR				CQ96BITPACKEN	NU	CQDATAWIDTH	
R/W-0h				R/W-0h	R-	R/W-0h	

Table 4-114. CQCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU3	R		
30-24	CQ2BASEADDR	R/W	40h	Address to be added to be internal address pointer for CQ2 (ADC/RxIF Saturation Detection)
23	NU2	R		
22-16	CQ1BASEADDR	R/W	20h	Address to be added to be internal address pointer for CQ1 (Signal Image Band Energy detection)
15-11	NU1	R		
10-4	CQ0BASEADDR	R/W	0h	Address to be added to be internal address pointer for CQ0 (Wide Band Energy detection)
3	CQ96BITPACKEN	R/W	0h	This is used to pack the CQ data into only the LSB 96 bits of each row of the CQ memory. This can be used in 3 channel mode of LVDS where the ADC data and Chirp Params occupy only LSB 96 bits of each memory row.
2	NU	R		
1-0	CQDATAWIDTH	R/W	0h	This is used to appropriately pack the valid CQ data bits in appropriate bits in the CQ memory. 00, 01->Raw 16, 10-->Raw 12, 11-->Raw14

4.6.4.7 TPCCPARSTATCFG Register (Offset = 80h) [reset = 0h]

TPCCPARSTATCFG is shown in Figure 4-888 and described in Table 4-931.

Return to Summary Table.

Figure 4-99. TPCCPARSTATCFG Register

31	30	29	28	27	26	25	24
NU							
R-							
23	22	21	20	19	18	17	16
NU							
R-							
15	14	13	12	11	10	9	8
NU					TPCCPARITYTSTEN	TPCCPARITYEN	TPCCPARITYCLR
R-					R/W-0h	R/W-0h	0h
7	6	5	4	3	2	1	0
TPCCPARITYSTAT							
R-0h							

Table 4-115. TPCCPARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R		
10	TPCCPARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in TPCC
9	TPCCPARITYEN	R/W	0h	Enable bit for the Parity computation in TPCC
8	TPCCPARITYCLR		0h	Clear bit for the Parity error from TPCC
7-0	TPCCPARITYSTAT	R	0h	Parity address from TPCC

4.6.4.8 CSI2TXPARSTATCFG Register (Offset = 84h) [reset = 0h]

CSI2TXPARSTATCFG is shown in Figure 4-100 and described in Table 4-116.

Return to Summary Table.

Figure 4-100. CSI2TXPARSTATCFG Register

31	30	29	28	27	26	25	24
NU2							
R-							
23	22	21	20	19	18	17	16
NU2							
R-							
15	14	13	12	11	10	9	8
NU2					CSI2TXPARITYTSTEN	CSI2TXPARITYEN	CSI2TXPARITYCLR
R-					R/W-0h	R/W-0h	0h
7	6	5	4	3	2	1	0
NU1	CSI2TXPARITYSTAT						
R-	R-0h						

Table 4-116. CSI2TXPARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU2	R		
10	CSI2TXPARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in CSI2
9	CSI2TXPARITYEN	R/W	0h	Enable bit for the Parity computation in CSI2
8	CSI2TXPARITYCLR		0h	Clear bit for the Parity error from CSI2
7	NU1	R		
6-0	CSI2TXPARITYSTAT	R	0h	Parity address from CSI2

4.6.4.9 CSICFG1 Register (Offset = A0h) [reset = 00FBFE7Fh]

CSICFG1 is shown in Figure 4-101 and described in Table 4-117.

Return to Summary Table.

Figure 4-101. CSICFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
CSIPIPD4	CSIPIPD3	CSIPIPD2	CSIPIPD1	CSIPIPD0	RESERVED	CSISIDLEACK	CSILANEENABLE
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-0h	R-1h	R/W-1Fh
15	14	13	12	11	10	9	8
CSILANEENABLE				RESERVED		CSIMIDLEREQ	RESERVED
R/W-1Fh				R/W-3h		R/W-1h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED							
R/W-0h							

Table 4-117. CSICFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	CSIPIPD4	R/W	1h	connected to CSI2 Protocol port name "csi_pipd4"
22	CSIPIPD3	R/W	1h	connected to CSI2 Protocol port name "csi_pipd3"
21	CSIPIPD2	R/W	1h	connected to CSI2 Protocol port name "csi_pipd2"
20	CSIPIPD1	R/W	1h	connected to CSI2 Protocol port name "csi_pipd1"
19	CSIPIPD0	R/W	1h	connected to CSI2 Protocol port name "csi_pipd0"
18	RESERVED	R	0h	Reserved
17	CSISIDLEACK	R	1h	connected to CSI2 Protocol port name "csi_po_sidleack"
16-12	CSILANEENABLE	R/W	1Fh	connected to CSI2 Protocol port name "csi_laneenable"
11-10	RESERVED	R/W	3h	Reserved
9	CSIMIDLEREQ	R/W	1h	connected to CSI2 Protocol port name "csi_pi_midlereq"
8-0	RESERVED	R/W	0h	Reserved

4.6.4.10 TPTC0WRMPUSTADD0 Register (Offset = 104h) [reset = 0h]

TPTC0WRMPUSTADD0 is shown in Figure 4-889 and described in Table 4-932.

Return to Summary Table.

Figure 4-102. TPTC0WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD0																															
R/W-0h																															

Table 4-118. TPTC0WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD0	R/W	0h	Start address0 for the MPU on the write port of TPTC0

4.6.4.11 TPTC0WRMPUSTADD1 Register (Offset = 108h) [reset = 0h]

TPTC0WRMPUSTADD1 is shown in Figure 4-890 and described in Table 4-933.

Return to Summary Table.

Figure 4-103. TPTC0WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD1																															
R/W-0h																															

Table 4-119. TPTC0WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD1	R/W	0h	Start address1 for the MPU on the write port of TPTC0

4.6.4.12 TPTC0WRMPUSTADD2 Register (Offset = 10Ch) [reset = 0h]

TPTC0WRMPUSTADD2 is shown in Figure 4-891 and described in Table 4-934.

Return to Summary Table.

Figure 4-104. TPTC0WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD2																															
R/W-0h																															

Table 4-120. TPTC0WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD2	R/W	0h	Start address2 for the MPU on the write port of TPTC0

4.6.4.13 TPTC0WRMPUSTADD3 Register (Offset = 110h) [reset = 0h]

TPTC0WRMPUSTADD3 is shown in Figure 4-892 and described in Table 4-935.

Return to Summary Table.

Figure 4-105. TPTC0WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD3																															
R/W-0h																															

Table 4-121. TPTC0WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD3	R/W	0h	Start address3 for the MPU on the write port of TPTC0

4.6.4.14 TPTC0WRMPUSTADD4 Register (Offset = 114h) [reset = 0h]

TPTC0WRMPUSTADD4 is shown in Figure 4-893 and described in Table 4-936.

Return to Summary Table.

Figure 4-106. TPTC0WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD4																															
R/W-0h																															

Table 4-122. TPTC0WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD4	R/W	0h	Start address4 for the MPU on the write port of TPTC0

4.6.4.15 TPTC0WRMPUSTADD5 Register (Offset = 118h) [reset = 0h]

TPTC0WRMPUSTADD5 is shown in Figure 4-894 and described in Table 4-937.

Return to Summary Table.

Figure 4-107. TPTC0WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD5																															
R/W-0h																															

Table 4-123. TPTC0WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD5	R/W	0h	Start address5 for the MPU on the write port of TPTC0

4.6.4.16 TPTC0WRMPUSTADD6 Register (Offset = 11Ch) [reset = 0h]

TPTC0WRMPUSTADD6 is shown in Figure 4-108 and described in Table 4-124.

Return to Summary Table.

Figure 4-108. TPTC0WRMPUSTADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD6																															
R/W-0h																															

Table 4-124. TPTC0WRMPUSTADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD6	R/W	0h	Start address6 for the MPU on the write port of TPTC0

4.6.4.17 TPTC0WRMPUSTADD7 Register (Offset = 120h) [reset = 0h]

TPTC0WRMPUSTADD7 is shown in Figure 4-109 and described in Table 4-125.

Return to Summary Table.

Figure 4-109. TPTC0WRMPUSTADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD7																															
R/W-0h																															

Table 4-125. TPTC0WRMPUSTADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD7	R/W	0h	Start address7 for the MPU on the write port of TPTC0

4.6.4.18 TPTC0WRMPUENDADD0 Register (Offset = 124h) [reset = 0h]

TPTC0WRMPUENDADD0 is shown in Figure 4-895 and described in Table 4-938.

Return to Summary Table.

Figure 4-110. TPTC0WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD0																															
R/W-0h																															

Table 4-126. TPTC0WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD0	R/W	0h	End address0 for the MPU on the write port of TPTC0

4.6.4.19 TPTC0WRMPUENDADD1 Register (Offset = 128h) [reset = 0h]

TPTC0WRMPUENDADD1 is shown in Figure 4-896 and described in Table 4-939.

Return to Summary Table.

Figure 4-111. TPTC0WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD1																															
R/W-0h																															

Table 4-127. TPTC0WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD1	R/W	0h	End address1 for the MPU on the write port of TPTC0

4.6.4.20 TPTC0WRMPUENDADD2 Register (Offset = 12Ch) [reset = 0h]

TPTC0WRMPUENDADD2 is shown in Figure 4-897 and described in Table 4-940.

Return to Summary Table.

Figure 4-112. TPTC0WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD2																															
R/W-0h																															

Table 4-128. TPTC0WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD2	R/W	0h	End address2 for the MPU on the write port of TPTC0

4.6.4.21 TPTC0WRMPUENDADD3 Register (Offset = 130h) [reset = 0h]

TPTC0WRMPUENDADD3 is shown in Figure 4-898 and described in Table 4-941.

Return to Summary Table.

Figure 4-113. TPTC0WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD3																															
R/W-0h																															

Table 4-129. TPTC0WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD3	R/W	0h	End address3 for the MPU on the write port of TPTC0

4.6.4.22 TPTC0WRMPUENDADD4 Register (Offset = 134h) [reset = 0h]

TPTC0WRMPUENDADD4 is shown in Figure 4-899 and described in Table 4-942.

Return to Summary Table.

Figure 4-114. TPTC0WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD4																															
R/W-0h																															

Table 4-130. TPTC0WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD4	R/W	0h	End address4 for the MPU on the write port of TPTC0

4.6.4.23 TPTC0WRMPUENDADD5 Register (Offset = 138h) [reset = 0h]

TPTC0WRMPUENDADD5 is shown in Figure 4-900 and described in Table 4-943.

Return to Summary Table.

Figure 4-115. TPTC0WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD5																															
R/W-0h																															

Table 4-131. TPTC0WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD5	R/W	0h	End address5 for the MPU on the write port of TPTC0

4.6.4.24 TPTC0WRMPUENDADD6 Register (Offset = 13Ch) [reset = 0h]

TPTC0WRMPUENDADD6 is shown in Figure 4-116 and described in Table 4-132.

Return to Summary Table.

Figure 4-116. TPTC0WRMPUENDADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD6																															
R/W-0h																															

Table 4-132. TPTC0WRMPUENDADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD6	R/W	0h	End address6 for the MPU on the write port of TPTC0

4.6.4.25 TPTC0WRMPUENDADD7 Register (Offset = 140h) [reset = 0h]

TPTC0WRMPUENDADD7 is shown in Figure 4-117 and described in Table 4-133.

Return to Summary Table.

Figure 4-117. TPTC0WRMPUENDADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD7																															
R/W-0h																															

Table 4-133. TPTC0WRMPUENDADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD7	R/W	0h	End address7 for the MPU on the write port of TPTC0

4.6.4.26 TPTC0WRMPUERRADD Register (Offset = 144h) [reset = 0h]

TPTC0WRMPUERRADD is shown in Figure 4-901 and described in Table 4-944.

Return to Summary Table.

Figure 4-118. TPTC0WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUERRADD																															
R-0h																															

Table 4-134. TPTC0WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUERRADD	R	0h	Error address for the MPU on the write port of TPTC0

4.6.4.27 TPTC0RDMPUSTADD0 Register (Offset = 148h) [reset = 0h]

TPTC0RDMPUSTADD0 is shown in Figure 4-902 and described in Table 4-945.

Return to Summary Table.

Figure 4-119. TPTC0RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD0																															
R/W-0h																															

Table 4-135. TPTC0RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD0	R/W	0h	Start address0 for the MPU on the read port of TPTC0

4.6.4.28 TPTC0RDMPUSTADD1 Register (Offset = 14Ch) [reset = 0h]

TPTC0RDMPUSTADD1 is shown in Figure 4-903 and described in Table 4-946.

Return to Summary Table.

Figure 4-120. TPTC0RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD1																															
R/W-0h																															

Table 4-136. TPTC0RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD1	R/W	0h	Start address1 for the MPU on the read port of TPTC0

4.6.4.29 TPTC0RDMPUSTADD2 Register (Offset = 150h) [reset = 0h]

TPTC0RDMPUSTADD2 is shown in Figure 4-904 and described in Table 4-947.

Return to Summary Table.

Figure 4-121. TPTC0RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD2																															
R/W-0h																															

Table 4-137. TPTC0RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD2	R/W	0h	Start address2 for the MPU on the read port of TPTC0

4.6.4.30 TPTC0RDMPUSTADD3 Register (Offset = 154h) [reset = 0h]

TPTC0RDMPUSTADD3 is shown in Figure 4-905 and described in Table 4-948.

Return to Summary Table.

Figure 4-122. TPTC0RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD3																															
R/W-0h																															

Table 4-138. TPTC0RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD3	R/W	0h	Start address3 for the MPU on the read port of TPTC0

4.6.4.31 TPTC0RDMPUSTADD4 Register (Offset = 158h) [reset = 0h]

TPTC0RDMPUSTADD4 is shown in Figure 4-906 and described in Table 4-949.

Return to Summary Table.

Figure 4-123. TPTC0RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD4																															
R/W-0h																															

Table 4-139. TPTC0RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD4	R/W	0h	Start address4 for the MPU on the read port of TPTC0

4.6.4.32 TPTC0RDMPUSTADD5 Register (Offset = 15Ch) [reset = 0h]

TPTC0RDMPUSTADD5 is shown in Figure 4-907 and described in Table 4-950.

Return to Summary Table.

Figure 4-124. TPTC0RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD5																															
R/W-0h																															

Table 4-140. TPTC0RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD5	R/W	0h	Start address5 for the MPU on the read port of TPTC0

4.6.4.33 TPTC0RDMPUSTADD6 Register (Offset = 160h) [reset = 0h]

TPTC0RDMPUSTADD6 is shown in Figure 4-125 and described in Table 4-141.

Return to Summary Table.

Figure 4-125. TPTC0RDMPUSTADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD6																															
R/W-0h																															

Table 4-141. TPTC0RDMPUSTADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD6	R/W	0h	Start address6 for the MPU on the read port of TPTC0

4.6.4.34 TPTC0RDMPUSTADD7 Register (Offset = 164h) [reset = 0h]

TPTC0RDMPUSTADD7 is shown in Figure 4-126 and described in Table 4-142.

Return to Summary Table.

Figure 4-126. TPTC0RDMPUSTADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD7																															
R/W-0h																															

Table 4-142. TPTC0RDMPUSTADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD7	R/W	0h	Start address7 for the MPU on the read port of TPTC0

4.6.4.35 TPTC0RDMPUENDADD0 Register (Offset = 168h) [reset = 0h]

TPTC0RDMPUENDADD0 is shown in Figure 4-908 and described in Table 4-951.

Return to Summary Table.

Figure 4-127. TPTC0RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD0																															
R/W-0h																															

Table 4-143. TPTC0RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD0	R/W	0h	End address0 for the MPU on the read port of TPTC0

4.6.4.36 TPTC0RDMPUENDADD1 Register (Offset = 16Ch) [reset = 0h]

TPTC0RDMPUENDADD1 is shown in Figure 4-909 and described in Table 4-952.

Return to Summary Table.

Figure 4-128. TPTC0RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD1																															
R/W-0h																															

Table 4-144. TPTC0RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD1	R/W	0h	End address1 for the MPU on the read port of TPTC0

4.6.4.37 TPTC0RDMPUENDADD2 Register (Offset = 170h) [reset = 0h]

TPTC0RDMPUENDADD2 is shown in Figure 4-910 and described in Table 4-953.

Return to Summary Table.

Figure 4-129. TPTC0RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD2																															
R/W-0h																															

Table 4-145. TPTC0RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD2	R/W	0h	End address2 for the MPU on the read port of TPTC0

4.6.4.38 TPTC0RDMPUENDADD3 Register (Offset = 174h) [reset = 0h]

TPTC0RDMPUENDADD3 is shown in Figure 4-911 and described in Table 4-954.

Return to Summary Table.

Figure 4-130. TPTC0RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD3																															
R/W-0h																															

Table 4-146. TPTC0RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD3	R/W	0h	End address3 for the MPU on the read port of TPTC0

4.6.4.39 TPTC0RDMPUENDADD4 Register (Offset = 178h) [reset = 0h]

TPTC0RDMPUENDADD4 is shown in Figure 4-912 and described in Table 4-955.

Return to Summary Table.

Figure 4-131. TPTC0RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD4																															
R/W-0h																															

Table 4-147. TPTC0RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD4	R/W	0h	End address4 for the MPU on the read port of TPTC0

4.6.4.40 TPTC0RDMPUENDADD5 Register (Offset = 17Ch) [reset = 0h]

TPTC0RDMPUENDADD5 is shown in Figure 4-913 and described in Table 4-956.

Return to Summary Table.

Figure 4-132. TPTC0RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD5																															
R/W-0h																															

Table 4-148. TPTC0RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD5	R/W	0h	End address5 for the MPU on the read port of TPTC0

4.6.4.41 TPTC0RDMPUENDADD6 Register (Offset = 180h) [reset = 0h]

TPTC0RDMPUENDADD6 is shown in Figure 4-133 and described in Table 4-149.

Return to Summary Table.

Figure 4-133. TPTC0RDMPUENDADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD6																															
R/W-0h																															

Table 4-149. TPTC0RDMPUENDADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD6	R/W	0h	End address6 for the MPU on the read port of TPTC0

4.6.4.42 TPTC0RDMPUENDADD7 Register (Offset = 184h) [reset = 0h]

TPTC0RDMPUENDADD7 is shown in Figure 4-134 and described in Table 4-150.

Return to Summary Table.

Figure 4-134. TPTC0RDMPUENDADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD7																															
R/W-0h																															

Table 4-150. TPTC0RDMPUENDADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD7	R/W	0h	End address7 for the MPU on the read port of TPTC0

4.6.4.43 TPTC0RDMPUERRADD Register (Offset = 188h) [reset = 0h]

TPTC0RDMPUERRADD is shown in Figure 4-914 and described in Table 4-957.

Return to Summary Table.

Figure 4-135. TPTC0RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUERRADD																															
R-0h																															

Table 4-151. TPTC0RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUERRADD	R	0h	Error address for the MPU on the read port of TPTC0

4.6.4.44 TPTC1WRMPUSTADD0 Register (Offset = 18Ch) [reset = 0h]

TPTC1WRMPUSTADD0 is shown in Figure 4-915 and described in Table 4-958.

Return to Summary Table.

Figure 4-136. TPTC1WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD0																															
R/W-0h																															

Table 4-152. TPTC1WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD0	R/W	0h	Start address0 for the MPU on the write port of TPTC1

4.6.4.45 TPTC1WRMPUSTADD1 Register (Offset = 190h) [reset = 0h]

TPTC1WRMPUSTADD1 is shown in Figure 4-916 and described in Table 4-959.

Return to Summary Table.

Figure 4-137. TPTC1WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD1																															
R/W-0h																															

Table 4-153. TPTC1WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD1	R/W	0h	Start address1 for the MPU on the write port of TPTC1

4.6.4.46 TPTC1WRMPUSTADD2 Register (Offset = 194h) [reset = 0h]

TPTC1WRMPUSTADD2 is shown in Figure 4-917 and described in Table 4-960.

Return to Summary Table.

Figure 4-138. TPTC1WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD2																															
R/W-0h																															

Table 4-154. TPTC1WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD2	R/W	0h	Start address2 for the MPU on the write port of TPTC1

4.6.4.47 TPTC1WRMPUSTADD3 Register (Offset = 198h) [reset = 0h]

TPTC1WRMPUSTADD3 is shown in Figure 4-918 and described in Table 4-961.

Return to Summary Table.

Figure 4-139. TPTC1WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD3																															
R/W-0h																															

Table 4-155. TPTC1WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD3	R/W	0h	Start address3 for the MPU on the write port of TPTC1

4.6.4.48 TPTC1WRMPUSTADD4 Register (Offset = 19Ch) [reset = 0h]

TPTC1WRMPUSTADD4 is shown in Figure 4-919 and described in Table 4-962.

Return to Summary Table.

Figure 4-140. TPTC1WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD4																															
R/W-0h																															

Table 4-156. TPTC1WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD4	R/W	0h	Start address4 for the MPU on the write port of TPTC1

4.6.4.49 TPTC1WRMPUSTADD5 Register (Offset = 1A0h) [reset = 0h]

TPTC1WRMPUSTADD5 is shown in Figure 4-920 and described in Table 4-963.

Return to Summary Table.

Figure 4-141. TPTC1WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD5																															
R/W-0h																															

Table 4-157. TPTC1WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD5	R/W	0h	Start address5 for the MPU on the write port of TPTC1

4.6.4.50 TPTC1WRMPUSTADD6 Register (Offset = 1A4h) [reset = 0h]

TPTC1WRMPUSTADD6 is shown in Figure 4-142 and described in Table 4-158.

Return to Summary Table.

Figure 4-142. TPTC1WRMPUSTADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD6																															
R/W-0h																															

Table 4-158. TPTC1WRMPUSTADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD6	R/W	0h	Start address6 for the MPU on the write port of TPTC1

4.6.4.51 TPTC1WRMPUSTADD7 Register (Offset = 1A8h) [reset = 0h]

TPTC1WRMPUSTADD7 is shown in Figure 4-143 and described in Table 4-159.

Return to Summary Table.

Figure 4-143. TPTC1WRMPUSTADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD7																															
R/W-0h																															

Table 4-159. TPTC1WRMPUSTADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD7	R/W	0h	Start address7 for the MPU on the write port of TPTC1

4.6.4.52 TPTC1WRMPUENDADD0 Register (Offset = 1ACh) [reset = 0h]

TPTC1WRMPUENDADD0 is shown in Figure 4-921 and described in Table 4-964.

Return to Summary Table.

Figure 4-144. TPTC1WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD0																															
R/W-0h																															

Table 4-160. TPTC1WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD0	R/W	0h	End address0 for the MPU on the write port of TPTC1

4.6.4.53 TPTC1WRMPUENDADD1 Register (Offset = 1B0h) [reset = 0h]

TPTC1WRMPUENDADD1 is shown in Figure 4-922 and described in Table 4-965.

Return to Summary Table.

Figure 4-145. TPTC1WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD1																															
R/W-0h																															

Table 4-161. TPTC1WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD1	R/W	0h	End address1 for the MPU on the write port of TPTC1

4.6.4.54 TPTC1WRMPUENDADD2 Register (Offset = 1B4h) [reset = 0h]

TPTC1WRMPUENDADD2 is shown in Figure 4-923 and described in Table 4-966.

Return to Summary Table.

Figure 4-146. TPTC1WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD2																															
R/W-0h																															

Table 4-162. TPTC1WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD2	R/W	0h	End address2 for the MPU on the write port of TPTC1

4.6.4.55 TPTC1WRMPUENDADD3 Register (Offset = 1B8h) [reset = 0h]

TPTC1WRMPUENDADD3 is shown in Figure 4-924 and described in Table 4-967.

Return to Summary Table.

Figure 4-147. TPTC1WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD3																															
R/W-0h																															

Table 4-163. TPTC1WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD3	R/W	0h	End address3 for the MPU on the write port of TPTC1

4.6.4.56 TPTC1WRMPUENDADD4 Register (Offset = 1BCh) [reset = 0h]

TPTC1WRMPUENDADD4 is shown in Figure 4-925 and described in Table 4-968.

Return to Summary Table.

Figure 4-148. TPTC1WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD4																															
R/W-0h																															

Table 4-164. TPTC1WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD4	R/W	0h	End address4 for the MPU on the write port of TPTC1

4.6.4.57 TPTC1WRMPUENDADD5 Register (Offset = 1C0h) [reset = 0h]

TPTC1WRMPUENDADD5 is shown in Figure 4-926 and described in Table 4-969.

Return to Summary Table.

Figure 4-149. TPTC1WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD5																															
R/W-0h																															

Table 4-165. TPTC1WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD5	R/W	0h	End address5 for the MPU on the write port of TPTC1

4.6.4.58 TPTC1WRMPUENDADD6 Register (Offset = 1C4h) [reset = 0h]

TPTC1WRMPUENDADD6 is shown in Figure 4-150 and described in Table 4-166.

Return to Summary Table.

Figure 4-150. TPTC1WRMPUENDADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD6																															
R/W-0h																															

Table 4-166. TPTC1WRMPUENDADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD6	R/W	0h	End address6 for the MPU on the write port of TPTC1

4.6.4.59 TPTC1WRMPUENDADD7 Register (Offset = 1C8h) [reset = 0h]

TPTC1WRMPUENDADD7 is shown in Figure 4-151 and described in Table 4-167.

Return to Summary Table.

Figure 4-151. TPTC1WRMPUENDADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD7																															
R/W-0h																															

Table 4-167. TPTC1WRMPUENDADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD7	R/W	0h	End address7 for the MPU on the write port of TPTC1

4.6.4.60 TPTC1WRMPUERRADD Register (Offset = 1CCh) [reset = 0h]

TPTC1WRMPUERRADD is shown in Figure 4-927 and described in Table 4-970.

Return to Summary Table.

Figure 4-152. TPTC1WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUERRADD																															
R-0h																															

Table 4-168. TPTC1WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUERRADD	R	0h	Error address for the MPU on the write port of TPTC1

4.6.4.61 TPTC1RDMPUSTADD0 Register (Offset = 1D0h) [reset = 0h]

TPTC1RDMPUSTADD0 is shown in Figure 4-928 and described in Table 4-971.

Return to Summary Table.

Figure 4-153. TPTC1RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD0																															
R/W-0h																															

Table 4-169. TPTC1RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD0	R/W	0h	Start address0 for the MPU on the read port of TPTC1

4.6.4.62 TPTC1RDMPUSTADD1 Register (Offset = 1D4h) [reset = 0h]

TPTC1RDMPUSTADD1 is shown in Figure 4-929 and described in Table 4-972.

Return to Summary Table.

Figure 4-154. TPTC1RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD1																															
R/W-0h																															

Table 4-170. TPTC1RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD1	R/W	0h	Start address1 for the MPU on the read port of TPTC1

4.6.4.63 TPTC1RDMPUSTADD2 Register (Offset = 1D8h) [reset = 0h]

TPTC1RDMPUSTADD2 is shown in Figure 4-930 and described in Table 4-973.

Return to Summary Table.

Figure 4-155. TPTC1RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD2																															
R/W-0h																															

Table 4-171. TPTC1RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD2	R/W	0h	Start address2 for the MPU on the read port of TPTC1

4.6.4.64 TPTC1RDMPUSTADD3 Register (Offset = 1DCh) [reset = 0h]

TPTC1RDMPUSTADD3 is shown in Figure 4-931 and described in Table 4-974.

Return to Summary Table.

Figure 4-156. TPTC1RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD3																															
R/W-0h																															

Table 4-172. TPTC1RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD3	R/W	0h	Start address3 for the MPU on the read port of TPTC1

4.6.4.65 TPTC1RDMPUSTADD4 Register (Offset = 1E0h) [reset = 0h]

TPTC1RDMPUSTADD4 is shown in Figure 4-932 and described in Table 4-975.

Return to Summary Table.

Figure 4-157. TPTC1RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD4																															
R/W-0h																															

Table 4-173. TPTC1RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD4	R/W	0h	Start address4 for the MPU on the read port of TPTC1

4.6.4.66 TPTC1RDMPUSTADD5 Register (Offset = 1E4h) [reset = 0h]

TPTC1RDMPUSTADD5 is shown in Figure 4-933 and described in Table 4-976.

Return to Summary Table.

Figure 4-158. TPTC1RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD5																															
R/W-0h																															

Table 4-174. TPTC1RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD5	R/W	0h	Start address5 for the MPU on the read port of TPTC1

4.6.4.67 TPTC1RDMPUSTADD6 Register (Offset = 1E8h) [reset = 0h]

TPTC1RDMPUSTADD6 is shown in Figure 4-159 and described in Table 4-175.

Return to Summary Table.

Figure 4-159. TPTC1RDMPUSTADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD6																															
R/W-0h																															

Table 4-175. TPTC1RDMPUSTADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD6	R/W	0h	Start address6 for the MPU on the read port of TPTC1

4.6.4.68 TPTC1RDMPUSTADD7 Register (Offset = 1ECh) [reset = 0h]

TPTC1RDMPUSTADD7 is shown in Figure 4-160 and described in Table 4-176.

Return to Summary Table.

Figure 4-160. TPTC1RDMPUSTADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD7																															
R/W-0h																															

Table 4-176. TPTC1RDMPUSTADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD7	R/W	0h	Start address7 for the MPU on the read port of TPTC1

4.6.4.69 TPTC1RDMPUENDADD0 Register (Offset = 1F0h) [reset = 0h]

TPTC1RDMPUENDADD0 is shown in Figure 4-934 and described in Table 4-977.

Return to Summary Table.

Figure 4-161. TPTC1RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD0																															
R/W-0h																															

Table 4-177. TPTC1RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD0	R/W	0h	End address0 for the MPU on the read port of TPTC1

4.6.4.70 TPTC1RDMPUENDADD1 Register (Offset = 1F4h) [reset = 0h]

TPTC1RDMPUENDADD1 is shown in Figure 4-935 and described in Table 4-978.

Return to Summary Table.

Figure 4-162. TPTC1RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD1																															
R/W-0h																															

Table 4-178. TPTC1RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD1	R/W	0h	End address1 for the MPU on the read port of TPTC1

4.6.4.71 TPTC1RDMPUENDADD2 Register (Offset = 1F8h) [reset = 0h]

TPTC1RDMPUENDADD2 is shown in Figure 4-936 and described in Table 4-979.

Return to Summary Table.

Figure 4-163. TPTC1RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD2																															
R/W-0h																															

Table 4-179. TPTC1RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD2	R/W	0h	End address2 for the MPU on the read port of TPTC1

4.6.4.72 TPTC1RDMPUENDADD3 Register (Offset = 1FCh) [reset = 0h]

TPTC1RDMPUENDADD3 is shown in Figure 4-937 and described in Table 4-980.

Return to Summary Table.

Figure 4-164. TPTC1RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD3																															
R/W-0h																															

Table 4-180. TPTC1RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD3	R/W	0h	End address3 for the MPU on the read port of TPTC1

4.6.4.73 TPTC1RDMPUENDADD4 Register (Offset = 200h) [reset = 0h]

TPTC1RDMPUENDADD4 is shown in Figure 4-938 and described in Table 4-981.

Return to Summary Table.

Figure 4-165. TPTC1RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD4																															
R/W-0h																															

Table 4-181. TPTC1RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD4	R/W	0h	End address4 for the MPU on the read port of TPTC1

4.6.4.74 TPTC1RDMPUENDADD5 Register (Offset = 204h) [reset = 0h]

TPTC1RDMPUENDADD5 is shown in Figure 4-939 and described in Table 4-982.

Return to Summary Table.

Figure 4-166. TPTC1RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD5																															
R/W-0h																															

Table 4-182. TPTC1RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD5	R/W	0h	End address5 for the MPU on the read port of TPTC1

4.6.4.75 TPTC1RDMPUENDADD6 Register (Offset = 208h) [reset = 0h]

TPTC1RDMPUENDADD6 is shown in Figure 4-167 and described in Table 4-183.

Return to Summary Table.

Figure 4-167. TPTC1RDMPUENDADD6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD6																															
R/W-0h																															

Table 4-183. TPTC1RDMPUENDADD6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD6	R/W	0h	End address6 for the MPU on the read port of TPTC1

4.6.4.76 TPTC1RDMPUENDADD7 Register (Offset = 20Ch) [reset = 0h]

TPTC1RDMPUENDADD7 is shown in Figure 4-168 and described in Table 4-184.

Return to Summary Table.

Figure 4-168. TPTC1RDMPUENDADD7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD7																															
R/W-0h																															

Table 4-184. TPTC1RDMPUENDADD7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD7	R/W	0h	End address7 for the MPU on the read port of TPTC1

4.6.4.77 TPTC1RDMPUERRADD Register (Offset = 210h) [reset = 0h]

TPTC1RDMPUERRADD is shown in Figure 4-940 and described in Table 4-983.

Return to Summary Table.

Figure 4-169. TPTC1RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUERRADD																															
R-0h																															

Table 4-185. TPTC1RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUERRADD	R	0h	Error address for the MPU on the read port of TPTC1

4.6.4.78 TPTCMPUVALIDCFG Register (Offset = 214h) [reset = 0h]

TPTCMPUVALIDCFG is shown in Figure 4-941 and described in Table 4-984.

Return to Summary Table.

Figure 4-170. TPTCMPUVALIDCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPTC1RDMPURNGVLD								TPTC1WRMPURNGVLD							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPURNGVLD								TPTC0WRMPURNGVLD							
R/W-0h								R/W-0h							

Table 4-186. TPTCMPUVALIDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TPTC1RDMPURNGVLD	R/W	0h	Valid bit for each address range for the MPU in the read port of TPTC1. [24]->Address0 and [31]-->Address7
23-16	TPTC1WRMPURNGVLD	R/W	0h	Valid bit for each address range for the MPU in the write port of TPTC1. [16]->Address0 and [23]-->Address7
15-8	TPTC0RDMPURNGVLD	R/W	0h	Valid bit for each address range for the MPU in the read port of TPTC0. [8]->Address0 and [15]-->Address7
7-0	TPTC0WRMPURNGVLD	R/W	0h	Valid bit for each address range for the MPU in the write port of TPTC0. [0]->Address0 and [7]-->Address7

4.6.4.79 TPTCMPUENCFG Register (Offset = 218h) [reset = 0h]

TPTCMPUENCFG is shown in Figure 4-942 and described in Table 4-985.

Return to Summary Table.

Figure 4-171. TPTCMPUENCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-							
23	22	21	20	19	18	17	16
RESERVED							
R-							
15	14	13	12	11	10	9	8
RESERVED							
R-							
7	6	5	4	3	2	1	0
TPTC1RDMPUERRCLR	TPTC1WRMPUERRCLR	TPTC0RDMPUERRCLR	TPTC0WRMPUERRCLR	TPTC1RDMPUEN	TPTC1WRMPUEN	TPTC0RDMPUEN	TPTC0WRMPUEN
0h	0h	0h	0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-187. TPTCMPUENCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R		Reserved
7	TPTC1RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC1.
6	TPTC1WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC1.
5	TPTC0RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC0.
4	TPTC0WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC0.
3	TPTC1RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC1.
2	TPTC1WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC1.
1	TPTC0RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC0.
0	TPTC0WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC0.

4.6.4.80 TESTPATTERNRX1ICFG Register (Offset = 21Ch) [reset = 00010000h]

TESTPATTERNRX1ICFG is shown in Figure 4-943 and described in Table 4-986.

Return to Summary Table.

Figure 4-172. TESTPATTERNRX1ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1IINCR																TSTPATRX1IOFFSET															
R/W-1h																R/W-0h															

Table 4-188. TESTPATTERNRX1ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.6.4.81 TESTPATTERNRX2ICFG Register (Offset = 220h) [reset = 00010000h]

TESTPATTERNRX2ICFG is shown in Figure 4-944 and described in Table 4-987.

Return to Summary Table.

Figure 4-173. TESTPATTERNRX2ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2IINCR																TSTPATRX2IOFFSET															
R/W-1h																R/W-0h															

Table 4-189. TESTPATTERNRX2ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 1.
15-0	TSTPATRX2IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 1.

4.6.4.82 TESTPATTERNRX3ICFG Register (Offset = 224h) [reset = 00010000h]

TESTPATTERNRX3ICFG is shown in Figure 4-945 and described in Table 4-988.

Return to Summary Table.

Figure 4-174. TESTPATTERNRX3ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3IINCR																TSTPATRX3IOFFSET															
R/W-1h																R/W-0h															

Table 4-190. TESTPATTERNRX3ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 2
15-0	TSTPATRX3IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 2

4.6.4.83 TESTPATTERNRX4ICFG Register (Offset = 228h) [reset = 00010000h]

TESTPATTERNRX4ICFG is shown in Figure 4-946 and described in Table 4-989.

Return to Summary Table.

Figure 4-175. TESTPATTERNRX4ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4IINCR																TSTPATRX4IOFFSET															
R/W-1h																R/W-0h															

Table 4-191. TESTPATTERNRX4ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 3
15-0	TSTPATRX4IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 3

4.6.4.84 TESTPATTERNRX1QCFG Register (Offset = 22Ch) [reset = 00010000h]

TESTPATTERNRX1QCFG is shown in Figure 4-947 and described in Table 4-990.

Return to Summary Table.

Figure 4-176. TESTPATTERNRX1QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1QINCR																TSTPATRX1QOFFSET															
R/W-1h																R/W-0h															

Table 4-192. TESTPATTERNRX1QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.6.4.85 TESTPATTERNRX2QCFG Register (Offset = 230h) [reset = 00010000h]

TESTPATTERNRX2QCFG is shown in Figure 4-948 and described in Table 4-991.

Return to Summary Table.

Figure 4-177. TESTPATTERNRX2QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2QINCR																TSTPATRX2QOFFSET															
R/W-1h																R/W-0h															

Table 4-193. TESTPATTERNRX2QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 1.
15-0	TSTPATRX2QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 1.

4.6.4.86 TESTPATTERNRX3QCFG Register (Offset = 234h) [reset = 00010000h]

TESTPATTERNRX3QCFG is shown in Figure 4-949 and described in Table 4-992.

Return to Summary Table.

Figure 4-178. TESTPATTERNRX3QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3QINCR																TSTPATRX3QOFFSET															
R/W-1h																R/W-0h															

Table 4-194. TESTPATTERNRX3QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 2
15-0	TSTPATRX3QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 2

4.6.4.87 TESTPATTERNRX4QCFG Register (Offset = 238h) [reset = 00010000h]

TESTPATTERNRX4QCFG is shown in Figure 4-950 and described in Table 4-993.

Return to Summary Table.

Figure 4-179. TESTPATTERNRX4QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4QINCR																TSTPATRX4QOFFSET															
R/W-1h																R/W-0h															

Table 4-195. TESTPATTERNRX4QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 3
15-0	TSTPATRX4QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 3

4.6.4.88 TESTPATTERNVLDCFG Register (Offset = 23Ch) [reset = 8h]

TESTPATTERNVLDCFG is shown in Figure 4-951 and described in Table 4-994.

Return to Summary Table.

Figure 4-180. TESTPATTERNVLDCFG Register

31	30	29	28	27	26	25	24
NU							
R-							
23	22	21	20	19	18	17	16
NU							
R-							
15	14	13	12	11	10	9	8
NU					TSTPATGENEN		
R-					R/W-0h		
7	6	5	4	3	2	1	0
TSTPATVLDCNT							
R/W-8h							

Table 4-196. TESTPATTERNVLDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R		
10-8	TSTPATGENEN	R/W	0h	Enable for test pattern generator. This is used to Mux with the functional data from BSS. 000 -->Disable, 111-->Enable, Others are reserved.
7-0	TSTPATVLDCNT	R/W	8h	Number of DSS Interconnect clocks (200 MHz) between successive samples for the test pattern gen.

4.6.4.89 DSSMISC Register (Offset = 240h) [reset = 0h]

DSSMISC is shown in Figure 4-181 and described in Table 4-197.

Return to Summary Table.

Figure 4-181. DSSMISC Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							FFTACCSLVEN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
FFTACCSLVEN		RESERVED					
R/W-0h		R/W-0h					

Table 4-197. DSSMISC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	Reserved
8-6	FFTACCSLVEN	R/W	0h	Enable for FFT Accelerator slaves (FFT_ACC_DMA1, DMA2 in VBUSM Interconnect and FFT_ACC in VBUSP Interconnect. 000 -->Disable, 111-->Enable, Others are reserved.
5-0	RESERVED	R/W	0h	Reserved

4.6.4.90 DSSMISC2 Register (Offset = 244h) [reset = 03020100h]

DSSMISC2 is shown in Figure 4-182 and described in Table 4-198.

Return to Summary Table.

Figure 4-182. DSSMISC2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU4		MSSCFGRNG3						NU3		MSSCFGRNG2					
R-		R/W-3h						R-		R/W-2h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2		MSSCFGRNG1						NU1		MSSCFGRNG0					
R-		R/W-1h						R-		R/W-0h					

Table 4-198. DSSMISC2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU4	R		
29-24	MSSCFGRNG3	R/W	3h	This is used in place of MSB 6 bits of the address sent from DSS to MSS if the bits [27:26] of the address bits from the interconnect is 11.
23-22	NU3	R		
21-16	MSSCFGRNG2	R/W	2h	This is used in place of MSB 6 bits of the address sent from DSS to MSS if the bits [27:26] of the address bits from the interconnect is 10.
15-14	NU2	R		
13-8	MSSCFGRNG1	R/W	1h	This is used in place of MSB 6 bits of the address sent from DSS to MSS if the bits [27:26] of the address bits from the interconnect is 01.
7-6	NU1	R		
5-0	MSSCFGRNG0	R/W	0h	This is used in place of MSB 6 bits of the address sent from DSS to MSS if the bits [27:26] of the address bits from the interconnect is 00.

4.7 16xx Control Registers

4.7.1 MSS_TOPRCM Registers

Table 4-802 lists the memory-mapped registers for the MSS_TOPRCM. All register offset addresses not listed in Table 4-802 should be considered as reserved locations and the register contents should not be modified.

Table 4-199. MSS_TOPRCM Registers

Offset	Acronym	Register Name	Section
8h	BSSCTL	Control Signals to BSS	Section 4.9.1.1
Ch	DSSCTL	Control Signals to DSS	Section 4.9.1.2
10h	EXTCLKDIV	Clock divide value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.3
14h	EXTCLKSRCSEL	Clock source select value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.4
18h	EXTCLKCTL	Clock gate control for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.5
1Ch	SOFTSYSRST	Software triggered Warm Reset	Section 4.9.1.6
20h	WDRSTEN	Issue Warm reset upon MSS Watch dog reset	Section 4.9.1.7
24h	SYSRSTCAUSE	Reset cause register	Section 4.9.1.8
28h	SYSRSTCAUSECLR	Clear Reset Cause register	Section 4.9.1.9
34h	MISCCAPT	Capture required Status values across the chip.	Section 4.9.1.10
38h	DCDCCTL0	PMIC_CLKOUT dethering control	Section 4.9.1.11
3Ch	DCDCCTL1	PMIC_CLKOUT dethering control	Section 4.9.1.12
48h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.1.13
4Ch	LVDSPADCTL0	LVDS pad control	Section 4.9.1.14
50h	LVDSPADCTL1	LVDS pad control	Section 4.9.1.15
60h	DFTREG0		Section 4.9.1.16
64h	DFTREG1		Section 4.9.1.17
74h	DFTREG5		Section 4.9.1.18
DCh	MEMINITDONE	Memory Initialization done status for memories in BSS and DSS	Section 4.9.1.19
ECh	SPARE0_3	Spare registers 0 to 3	Section 4.9.1.20
FCh	MSS_SIGNATURE	MSS Signature register	Section 4.9.1.21
158h	GEMBOOTSTCEN		Section 4.9.1.22
178h	MISCCTL1	Miscellaneous Control Register	Section 4.9.1.23
180h	USERMODEEN2	USER_MODE_ACCESS_EN	Section 4.9.1.24
18Ch	SYSTICK		Section 4.9.1.25
1C4h	SECURECFGREG1		Section 4.9.1.26
1C8h	SECURECFGREG2		Section 4.9.1.27
1CCh	SECURECFGREG3		Section 4.9.1.28
1D0h	SECURECFGREG4		Section 4.9.1.29
1D4h	SECURERAMREG		Section 4.9.1.30
1E4h	SPAREMULTIBIT	MIBSPI setup register	Section 4.9.1.31
1E8h	SPARE4_9	Spare registers 4 to 9	Section 4.9.1.32
200h	UID31TO0	Efuse Row Capture register for FROM1	Section 4.9.1.33
204h	UID63TO32	Efuse Row Capture register for FROM1	Section 4.9.1.34
208h	UID95TO64	Efuse Row Capture register for FROM1	Section 4.9.1.35
20Ch	UID119TO96	Efuse Row Capture register for FROM1	Section 4.9.1.36
2A8h	MEMINITSTARTSHMEM	Shared memory initialization start	Section 4.9.1.37
2ACh	MEMINITDONESHMEM	Shared memory intialization end	Section 4.9.1.38
2B0h	DSSMEMTAB0	Controls ordering of banks in shared memory associated with DSS	Section 4.9.1.39
2BCh	TCMAMEMTAB	Controls ordering of banks in shared memory associated with MSS TCMA	Section 4.9.1.40
2C0h	TCMBMEMTAB	Controls ordering of banks in shared memory associated with MSS TCMB	Section 4.9.1.41
2C8h	SHMEMBANKSEL3TO0	Shared memory master allocation.	Section 4.9.1.42

Table 4-199. MSS_TOPRCM Registers (continued)

Offset	Acronym	Register Name	Section
2CCh	SHMEMBANKSEL7TO4	Shared memory master allocation.	Section 4.9.1.43
2D0h	PBISTCLKCTL	PBIST clock control register	Section 4.9.1.44

Complex bit access types are encoded to fit into small table cells. Table 4-803 shows the codes that are used for access types in this section.

Table 4-200. MSS_TOPRCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.7.1.1 BSSCTL Register (Offset = 8h) [reset = ADADADADh]

BSSCTL is shown in Figure 4-767 and described in Table 4-804.

Return to Summary Table.

Control Signals to BSS

Figure 4-183. BSSCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BSSCPUHALT								RESERVED							
R/W-ADh								R/W-ADh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
R/W-ADh															

Table 4-201. BSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BSSCPUHALT	R/W	ADh	Halt BSS CR4, To halt : either 3:0 should be 0xD or 7:4 should be 0xA One should Halt the processor before releasing BSS reset.
23-0	RESERVED	R/W	ADh	Reserved

4.7.1.2 DSSCTL Register (Offset = Ch) [reset = 00ADADADh]

DSSCTL is shown in Figure 4-768 and described in Table 4-805.

Return to Summary Table.

Control Signals to DSS

Figure 4-184. DSSCTL Register

31	30	29	28	27	26	25	24
NU					GEMLRSTN	GEMGRSTN	GEMPORZ
0h					R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-ADh							
15	14	13	12	11	10	9	8
RESERVED							
R/W-ADh							
7	6	5	4	3	2	1	0
RESERVED							
R/W-ADh							

Table 4-202. DSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU		0h	Reserved
26	GEMLRSTN	R/W	0h	DSP Local Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
25	GEMGRSTN	R/W	0h	DSP Global Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
24	GEMPORZ	R/W	0h	DSP Por Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
23-0	RESERVED	R/W	ADh	Reserved

4.7.1.3 EXTCLKDIV Register (Offset = 10h) [reset = 0h]

EXTCLKDIV is shown in Figure 4-769 and described in Table 4-806.

Return to Summary Table.

Clock divide value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-185. EXTCLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2DIV								EXTCLK1DIV							
R/W-0h								R/W-0h							

Table 4-203. EXTCLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Reserved
15-8	EXTCLK2DIV	R/W	0h	Divide value for clock source (selected by field EXTCLKSRCSEL in register CLKSRCSEL.) to PMIC_CLKOUT generation logic “0000_0000” => div1 “0000_0001” => div2 | | “1111_1111” => div256 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming EXTCLK2SRCSEL.
7-0	EXTCLK1DIV	R/W	0h	Divide value for MCU_CLKOUT (the one going out of chip) source clock selected by field EXTCLKSRCSEL in register CLKSRCSEL. “0000_0000” => div1 “0000_0001” => div2 | | “1111_1111” => div256 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming EXTCLK1SRCSEL.

4.7.1.4 EXTCLKSRCSEL Register (Offset = 14h) [reset = 0h]

EXTCLKSRCSEL is shown in Figure 4-770 and described in Table 4-807.

Return to Summary Table.

Clock source select value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-186. EXTCLKSRCSEL Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1				EXTCLK2SRCSEL			
0h				R/W-0h			
7	6	5	4	3	2	1	0
NU0				EXTCLK1SRCSEL			
0h				R/W-0h			

Table 4-204. EXTCLKSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU1		0h	Reserved
11-8	EXTCLK2SRCSEL	R/W	0h	Select clock source for PMIC_CLKOUT generation “000” => CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) “001” => RCCLK (10Mhz) “010” => 600Mhz PLL divided clock “011” => 240Mhz PLL divided clock “100” => RCCLK (10Mhz) “101” => RCCLK (10Mhz) “110” => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mhz) “111” => RCCLK
7-4	NU0		0h	Reserved
3-0	EXTCLK1SRCSEL	R/W	0h	Select clock source for MCU_CLKOUT “000” =>CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) “001” => RCCLK (10Mhz) “010” => 600Mhz PLL divided clock “011” => 240Mhz PLL divided clock “100” => RCCLK (10Mhz) “101” => RCCLK (10Mhz) “110” => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mhz) “111” => RCCLK

4.7.1.5 EXTCLKCTL Register (Offset = 18h) [reset = ADADh]

EXTCLKCTL is shown in Figure 4-771 and described in Table 4-808.

Return to Summary Table.

Clock gate control for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-187. EXTCLKCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2GATE								EXTCLK1GATE							
R/W-ADh								R/W-ADh							

Table 4-205. EXTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Reserved
15-8	EXTCLK2GATE	R/W	ADh	Pre clock divider gate for PMIC_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA
7-0	EXTCLK1GATE	R/W	ADh	Pre clock divider gate for MCU_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA

4.7.1.6 SOFTSYSRST Register (Offset = 1Ch) [reset = 0h]

SOFTSYSRST is shown in Figure 4-772 and described in Table 4-809.

Return to Summary Table.

Software triggered Warm Reset

Figure 4-188. SOFTSYSRST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SOFTSYSRST							
0h								R/W-0h							

Table 4-206. SOFTSYSRST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	SOFTSYSRST	R/W	0h	Write 0xAD to trigger warm reset to the chip.

4.7.1.7 WDRSTEN Register (Offset = 20h) [reset = 0h]

WDRSTEN is shown in Figure 4-773 and described in Table 4-810.

Return to Summary Table.

Issue Warm reset upon MSS Watch dog reset

Figure 4-189. WDRSTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								WDRSTEN							
0h																								R/W-0h							

Table 4-207. WDRSTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	WDRSTEN	R/W	0h	Write 0xAD to trigger warm reset to the chip upon MSS wdog reset.

4.7.1.8 SYSRSTCAUSE Register (Offset = 24h) [reset = 0h]

SYSRSTCAUSE is shown in Figure 4-774 and described in Table 4-811.

Return to Summary Table.

Reset cause register

Figure 4-190. SYSRSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU												SYSRSTCAUSE			
0h												R-0h			

Table 4-208. SYSRSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU		0h	Reserved
3-0	SYSRSTCAUSE	R	0h	Gives cause of chip reset “1001” : System out of NRESET “1010” : Warm reset because of MSS Wdog. “1100” : Warm reset because of Software trigger- SOFTSYSRST “1000” : External Warm Reset Note: The ROM Bootloader clears this register as part of its processing, so it always reads zero by the application or secondary bootloader. However, the ROM Bootloader saves the original value into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.7.1.9 SYSRSTCAUSECLR Register (Offset = 28h) [reset = 0h]

SYSRSTCAUSECLR is shown in Figure 4-775 and described in Table 4-812.

Return to Summary Table.

Clear Reset Cause register

Figure 4-191. SYSRSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SYSRSTCAUSECLR							
0h								0h							

Table 4-209. SYSRSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	SYSRSTCAUSECLR		0h	Write 0xAD to clear SYSRSTCAUSE..Self clearing

4.7.1.10 MISCCAPT Register (Offset = 34h) [reset = 0h]

MISCCAPT is shown in Figure 4-776 and described in Table 4-813.

Return to Summary Table.

Capture required Status values across the chip.

Figure 4-192. MISCCAPT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCAPT																															
R-0h																															

Table 4-210. MISCCAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCAPT	R	0h	0: No error Any other non-zero value: Error

4.7.1.11 DCDCCTL0 Register (Offset = 38h) [reset = 0h]

DCDCCTL0 is shown in Figure 4-777 and described in Table 4-814.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-193. DCDCCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCDCCTL0																															
R/W-0h																															

Table 4-211. DCDCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DCDCCTL0	R/W	0h	[26:0] PMIC_CLKOUT Frequency slope value.Unsigned

4.7.1.12 DCDCCTL1 Register (Offset = 3Ch) [reset = 0h]

DCDCCTL1 is shown in Figure 4-778 and described in Table 4-815.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-194. DCDCCTL1 Register

31	30	29	28	27	26	25	24
DCDCCTL1							
R/W-0h							
23	22	21	20	19	18	17	16
DCDCCTL1							
R/W-0h							
15	14	13	12	11	10	9	8
DCDCCTL1							
R/W-0h							
7	6	5	4	3	2	1	0
DCDCCTL1						DCDCLKEN	DCDCRST
R/W-0h						R/W-0h	R/W-0h

Table 4-212. DCDCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	DCDCCTL1	R/W	0h	PMIC_CLKOUT dithering control [8] : Frequency accumulation Mode . ‘0’ : Continous mode.’1’ : Staircase mode [9] : ‘1’ Enables dither [23:16] : Minimum frequency threshold [31:24] : Maximum frequency threshold
1	DCDCLKEN	R/W	0h	PMIC_CLKOUT Enable – Multi Bit
0	DCDCRST	R/W	0h	PMIC_CLKOUT dithering control block reset (active high) – Multi Bit

4.7.1.13 USERMODEEN Register (Offset = 48h) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-195. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-213. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space from 0x00 to 0XFF

4.7.1.14 LVDSPADCTL0 Register (Offset = 4Ch) [reset = 01010101h]

LVDSPADCTL0 is shown in Figure 4-780 and described in Table 4-817.

Return to Summary Table.

LVDS pad control

Figure 4-196. LVDSPADCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL0																															
R/W-01010101h																															

Table 4-214. LVDSPADCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL0	R/W	01010101h	0 : pwrdn Control for i_LVDSclk_io_cell 1 : lopwra Control for i_LVDSclk_io_cell 2 : lopwrb Control for i_LVDSclk_io_cell 3 : lpsel Control for i_LVDSclk_io_cell 4 : sub_lvds_en Control for i_LVDSclk_io_cell 5 : hiz_disable Control for i_LVDSclk_io_cell 6 : ext_res_en Control for i_LVDSclk_io_cell 7 : Reserved 8 : pwrdn Control for i_LVDS_tx0_io_cell 9 : lopwra Control for i_LVDS_tx0_io_cell 10 : lopwrb Control for i_LVDS_tx0_io_cell 11 : lpsel Control for i_LVDS_tx0_io_cell 12 : sub_lvds_en Control for i_LVDS_tx0_io_cell 13 : hiz_disable Control for i_LVDS_tx0_io_cell 14 : ext_res_en Control for i_LVDS_tx0_io_cell 15 : Reserved 16 : pwrdn Control for i_LVDS_tx1_io_cell 17 : lopwra Control for i_LVDS_tx1_io_cell 18 : lopwrb Control for i_LVDS_tx1_io_cell 19 : lpsel Control for i_LVDS_tx1_io_cell 20 : sub_lvds_en Control for i_LVDS_tx1_io_cell 21 : hiz_disable Control for i_LVDS_tx1_io_cell 22 : ext_res_en Control for i_LVDS_tx1_io_cell 23 -31 : Reserved

4.7.1.15 LVDSPADCTL1 Register (Offset = 50h) [reset = 101h]

LVDSPADCTL1 is shown in Figure 4-781 and described in Table 4-818.

Return to Summary Table.

LVDS pad control

Figure 4-197. LVDSPADCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL1																															
R/W-101h																															

Table 4-215. LVDSPADCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL1	R/W	101h	0- 7 : Reserved 8 : pwrdn Control for i_LVDSfrclk_io_cell 9 : lopwra Control for i_LVDSfrclk_io_cell 10 : lopwrb Control for i_LVDSfrclk_io_cell 11 : lpsel Control for i_LVDSfrclk_io_cell 12 : sub_lvds_en Control for i_LVDSfrclk_io_cell 13 : hiz_disable Control for i_LVDSfrclk_io_cell 14 : ext_res_en Control for i_LVDSfrclk_io_cell 15-23: Reserved 24 : pwrdn Control for lvds_bias_cell 25 : efuse_set Control for lvds_bias_cell

4.7.1.16 DFTREG0 Register (Offset = 60h) [reset = 0h]

DFTREG0 is shown in Figure 4-782 and described in Table 4-819.

Return to Summary Table.

Figure 4-198. DFTREG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG0																															
R/W-0h																															

Table 4-216. DFTREG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG0	R/W	0h	[3:0] : MSS PBIST SELFTEST KEY = 4'b1010 [4] : Reserved [5] : PBIST IP reset control. 1 = reset, 0 = out of reset. The PBIST_SELFTEST_KEY must be enabled to program this bit. [31:6] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.7.1.17 DFTREG1 Register (Offset = 64h) [reset = 0h]

DFTREG1 is shown in Figure 4-783 and described in Table 4-820.

Return to Summary Table.

Figure 4-199. DFTREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG1																															
R/W-0h																															

Table 4-217. DFTREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG1	R/W	0h	[31:0] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.7.1.18 DFTREG5 Register (Offset = 74h) [reset = 0h]

DFTREG5 is shown in Figure 4-784 and described in Table 4-821.

Return to Summary Table.

Figure 4-200. DFTREG5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG5																															
R/W-0h																															

Table 4-218. DFTREG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG5	R/W	0h	[0] : Reserved [4:1] DSP PBIST SELFTEST KEY = 4'b1010 [8:5]: Subsystem level memory self-test power clock gate enable controls [31:9] : Reserved

4.7.1.19 MEMINITDONE Register (Offset = DCh) [reset = 0h]

MEMINITDONE is shown in Figure 4-822 and described in Table 4-861.

Return to Summary Table.

Memory Initialization done status for memories in BSS and DSS

Figure 4-201. MEMINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED						BSSVIMMEM	RESERVED
0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED							
0h							

Table 4-219. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED		0h	Reserved
9	BSSVIMMEM	R	0h	Memory Initialization done status for BSS VIM memory
8-0	RESERVED	R	0h	Reserved

4.7.1.20 SPARE0_3 Register (Offset = ECh, F0h, F4h, F8h) [reset = 0h]

SPARE0_3 is shown in Figure 4-786 and described in Table 4-823.

Return to Summary Table.

Spare Register

Figure 4-202. SPARE0_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE0_3																															
R/W-0h																															

Table 4-220. SPARE0_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE0_3	R/W	0h	Four 32-bit registers that may be used by an application for any purpose.

4.7.1.21 MSS_SIGNATURE Register (Offset = FCh) [reset = 0BB5202Fh]

MSS_SIGNATURE is shown in Figure 4-787 and described in Table 4-824.

Return to Summary Table.

Spare Register

Figure 4-203. MSS_SIGNATURE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSS_SIGNATURE																															
R/W-0BB5202Fh																															

Table 4-221. MSS_SIGNATURE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSS_SIGNATURE	R/W	0BB5202Fh	

4.7.1.22 GEMBOOTSTCEN Register (Offset = 158h) [reset = 0h]

GEMBOOTSTCEN is shown in Figure 4-788 and described in Table 4-825.

Return to Summary Table.

Figure 4-204. GEMBOOTSTCEN Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							GEMBOOTSTCEN
R-0h							R/W-0h

Table 4-222. GEMBOOTSTCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	
0	GEMBOOTSTCEN	R/W	0h	'1' : Enable GEM STC during GEM power UP

4.7.1.23 MISCCTL1 Register (Offset = 178h) [reset = 0h]

MISCCTL1 is shown in Figure 4-789 and described in Table 4-826.

Return to Summary Table.

Miscellaneous Control Register

Figure 4-205. MISCCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCTL1																															
R/W-0h																															

Table 4-223. MISCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCTL1	R/W	0h	7:0 : Write 0xAD to enable Warm_resetn from external device in addition to internally generated warm reset 16:8 : Write 0xAD to take board level loop back clock for QSPI. 24:16 : Write 0xAD to external clock as QSPI baud clock source – needed for DFT IO char..

4.7.1.24 USERMODEEN2 Register (Offset = 180h) [reset = 0h]

USERMODEEN2 is shown in Figure 4-790 and described in Table 4-827.

Return to Summary Table.

Figure 4-206. USERMODEEN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN2																															
R/W-0h																															

Table 4-224. USERMODEEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN2	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space which are resettable only by Power on reset. i.e. from offset address 0x100 to 0x1FF

4.7.1.25 SYSTICK Register (Offset = 18Ch) [reset = 0h]

SYSTICK is shown in Figure 4-791 and described in Table 4-828.

Return to Summary Table.

Figure 4-207. SYSTICK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSTICK																															
R-0h																															

Table 4-225. SYSTICK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYSTICK	R	0h	Continuous counter running on 32Khz derived from RC clock.

4.7.1.26 SECURECFGREG1 Register (Offset = 1C4h) [reset = 00700777h]

SECURECFGREG1 is shown in Figure 4-792 and described in Table 4-829.

Return to Summary Table.

Figure 4-208. SECURECFGREG1 Register

31	30	29	28	27	26	25	24
RESERVED	JTAGFIREWALLEN			RESERVED			
0h	R/W-0h			R/W-0h			
23	22	21	20	19	18	17	16
RESERVED	SECURERAMFIREWALLEN			RESERVED	LOGGERFIREWALLEN		
0h	R/W-7h			0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	TRACEFIREWALLEN			RESERVED	CRYPTOFIREWALLEN		
0h	R/W-0h			0h	R/W-7h		
7	6	5	4	3	2	1	0
RESERVED	CUSTCEKFIREWALLEN			RESERVED			
0h	R/W-7h			R/W-0h			

Table 4-226. SECURECFGREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED		0h	
30-28	JTAGFIREWALLEN	R/W	0h	JTAG Firewall.Firewall Disabled for value "111" and enabled for rest
27-23	RESERVED		0h	
22-20	SECURERAMFIREWALLEN	R/W	7h	Set Secure RAM Firewall.Firewall Disabled for value "111" and enabled for rest
19	RESERVED		0h	
18-16	LOGGERFIREWALLEN	R/W	0h	Set Logger Firewall.Firewall Disabled for value "111" and enabled for rest
15	RESERVED		0h	
14-12	TRACEFIREWALLEN	R/W	0h	Set Trace Firewall. Firewall Disabled for value "111" and enabled for rest
11	RESERVED		0h	
10-8	CRYPTOFIREWALLEN	R/W	7h	Set Crypto Firewall. Firewall Disabled for value "111" and enabled for rest
7	RESERVED		0h	
6-4	CUSTCEKFIREWALLEN	R/W	7h	Set CEK1,CEK2 firewall.Firewall Disabled for value "111" and enabled for rest
3-0	RESERVED		0h	

4.7.1.27 SECURECFGREG2 Register (Offset = 1C8h) [reset = 0h]

SECURECFGREG2 is shown in Figure 4-793 and described in Table 4-830.

Return to Summary Table.

Figure 4-209. SECURECFGREG2 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED					CUSTKEYERASE		
0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					DMMFIREWALLEN		
0h					R/W-0h		

Table 4-227. SECURECFGREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED		0h	Reserved
10-8	CUSTKEYERASE	R/W	0h	Erase CEK1 ,CEK2,CPK Keys when value "111" is written
7-3	RESERVED		0h	Reserved
2-0	DMMFIREWALLEN	R/W	0h	DMM Firewall.Firewall Disabled for value "111" and enabled for rest

4.7.1.28 SECURECFGREG3 Register (Offset = 1CCh) [reset = 0h]

SECURECFGREG3 is shown in Figure 4-794 and described in Table 4-831.

Return to Summary Table.

Figure 4-210. SECURECFGREG3 Register

31	30	29	28	27	26	25	24
RESERVED	JTAGSTICKYBIT			RESERVED			
0h	R/W-0h			0h			
23	22	21	20	19	18	17	16
RESERVED	SECURERAMSTICKYBIT			RESERVED	TRACESTICKYBIT		
0h	R/W-0h			0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	CRYPTOSTICKYBIT			RESERVED	CUSTCEKSTICKYBIT		
0h	R/W-0h			0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					LOGGERSTICKYBIT		
0h					R/W-0h		

Table 4-228. SECURECFGREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED		0h	
30-28	JTAGSTICKYBIT	R/W	0h	JTAG Sticky Reg. Sticky reg is set when value "111 is written
27-23	RESERVED		0h	
22-20	SECURERAMSTICKYBIT	R/W	0h	Secure RAM Sticky Reg. Sticky reg is set when value "111 is written
19	RESERVED		0h	
18-16	TRACESTICKYBIT	R/W	0h	Trace Sticky Reg. Sticky reg is set when value "111 is written
15	RESERVED		0h	
14-12	CRYPTOSTICKYBIT	R/W	0h	Crypto Sticky Reg. Sticky reg is set when value "111 is written
11	RESERVED		0h	
10-8	CUSTCEKSTICKYBIT	R/W	0h	CEK1,CEK2 Sticky Reg for firewall. Sticky reg is set when value "111 is written
7-3	RESERVED		0h	
2-0	LOGGERSTICKYBIT	R/W	0h	Logger Sticky Reg. Sticky reg is set when value "111 is written

4.7.1.29 SECURECFGREG4 Register (Offset = 1D0h) [reset = 0h]

SECURECFGREG4 is shown in Figure 4-795 and described in Table 4-832.

Return to Summary Table.

Figure 4-211. SECURECFGREG4 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RESERVED					DMMSTICKYBIT		
0h					R/W-0h		

Table 4-229. SECURECFGREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED		0h	Reserved
2-0	DMMSTICKYBIT	R/W	0h	DMM Sticky Reg. Sticky reg is set when value "111 is written

4.7.1.30 SECURERAMREG Register (Offset = 1D4h) [reset = 0h]

SECURERAMREG is shown in Figure 4-796 and described in Table 4-833.

Return to Summary Table.

Figure 4-212. SECURERAMREG Register

31	30	29	28	27	26	25	24
NU4							SECURERAMKEY255
0h							R/W-0h
23	22	21	20	19	18	17	16
SECURERAMKEYIDX							
R/W-0h							
15	14	13	12	11	10	9	8
NU2							SECURERAMKEYRD
0h							0h
7	6	5	4	3	2	1	0
NU10							SECURERAMRDDONE
0h							R-0h

Table 4-230. SECURERAMREG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4		0h	
24	SECURERAMKEY255	R/W	0h	1: Secure RAM key bitwidth = 255 0:Secure RAM key bitwidth = 128
23-16	SECURERAMKEYIDX	R/W	0h	Index to Secure RAM
15-9	NU2		0h	
8	SECURERAMKEYRD		0h	1: Load key from secure RAM into register to be used by AES engine.Self clearing
7-1	NU10		0h	
0	SECURERAMRDDONE	R	0h	Secure RAM key loaded into register

4.7.1.31 SPAREMULTIBIT Register (Offset = 1E4h) [reset = FFFF0000h]

SPAREMULTIBIT is shown in Figure 4-797 and described in Table 4-834.

Return to Summary Table.

Spare Register

Figure 4-213. SPAREMULTIBIT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-1h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-1h							
15	14	13	12	11	10	9	8
RESERVED				SPAREMULTIBIT11	SPAREMULTIBIT10	SPAREMULTIBIT9	SPAREMULTIBIT8
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SPAREMULTIBIT7	SPAREMULTIBIT6	SPAREMULTIBIT5	SPAREMULTIBIT4	SPAREMULTIBIT3	SPAREMULTIBIT2	SPAREMULTIBIT1	SPAREMULTIBIT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-231. SPAREMULTIBIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	1h	Reserved
11	SPAREMULTIBIT11	R/W	0h	MIBSPIB : when set the TRIGGER's are un-gated only when chip-select is active
10	SPAREMULTIBIT10	R/W	0h	SPIB trigger source polarity select.'0' - Polarity 0,'1'-Polarity 1
9	SPAREMULTIBIT9	R/W	0h	SPIA trigger source polarity select.'0' - Polarity 0,'1'-Polarity 1
8	SPAREMULTIBIT8	R/W	0h	'1': MIBSPIB External chip select is overridden with the value of MIBSPIB CS polarity-slave mode
7	SPAREMULTIBIT7	R/W	0h	'1': MIBSPIA External chip select is overridden with the value of MIBSPIA CS polarity-slave mode
6	SPAREMULTIBIT6	R/W	0h	MIBSPIB CS Trigger SRC enable '1': Use CS as trigger source
5	SPAREMULTIBIT5	R/W	0h	MIBSPIB CS polarity-slave mode 1: Active high 0:Active low
4	SPAREMULTIBIT4	R/W	0h	MIBSPIB MISO OE_N Control based on Chip select(CS)-applicable in slave mode 1:MISO OEN controlled based on CS.When CS is inactive OE_N=1 0:MISO OEN controlled by IP
3	SPAREMULTIBIT3	R/W	0h	MIBSPIA :When set the TRIGGER's are un-gated only when chip-select is active.
2	SPAREMULTIBIT2	R/W	0h	MIBSPIA CS Trigger SRC enable-slave mode '1': Use CS as trigger source
1	SPAREMULTIBIT1	R/W	0h	MIBSPIA CS polarity-slave mode 1: Active high 0:Active low
0	SPAREMULTIBIT0	R/W	0h	MIBSPIA MISO OE_N Control based on Chip select(CS)-applicable in slave mode 1:MISO OEN controlled based on CS.When CS is inactive OE_N=1,else controlled by IP 0:MISO OEN controlled by IP

4.7.1.32 SPARE4_9 Register (Offset = 1E8h, 1ECh, … 1FCh) [reset = 0h]

SPARE4_9 is shown in Figure 4-798 and described in Table 4-835.

Return to Summary Table.

Figure 4-214. SPARE4_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID31TO0																															
R-0h																															

Table 4-232. SPARE4_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE4_9	R	0h	Six 32-bit registers that may be used by an application for any purpose. SPARE4 through SPARE8 are not used by the ROM Bootloader, and will survive a Warm Reset. SPARE9 is used by the ROM Bootloader to record RCM_RSTCAUSE and TOPRCM_SYSRSTCAUSE before they are modified by the Bootloader. SPARE 9 is written by the Bootloader thusly: SPARE9[3:0] - current TOPRCM reset cause SPARE9[11:4] - current MSS RCM reset cause SPARE9[15:12] - UNUSED SPARE9[19:16] - previous TOP RCM reset cause SPARE9[27:20] - previous MSS RCM reset cause SPARE9[31:38] - UNUSED

4.7.1.33 UID31TO0 Register (Offset = 200h) [reset = 0h]

UID31TO0 is shown in Figure 4-799 and described in Table 4-836.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-215. UID31TO0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID31TO0																															
R-0h																															

Table 4-233. UID31TO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID31TO0	R	0h	FROM1 Efuse Read UID[31:0]

4.7.1.34 UID63TO32 Register (Offset = 204h) [reset = 0h]

UID63TO32 is shown in Figure 4-800 and described in Table 4-837.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-216. UID63TO32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID63TO32																															
R-0h																															

Table 4-234. UID63TO32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID63TO32	R	0h	FROM1 Efuse Read UID[63:32]

4.7.1.35 UID95TO64 Register (Offset = 208h) [reset = 0h]

UID95TO64 is shown in Figure 4-801 and described in Table 4-838.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-217. UID95TO64 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID95TO64																															
R-0h																															

Table 4-235. UID95TO64 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID95TO64	R	0h	FROM1 Efuse Read UID[95:64]

4.7.1.36 UID119TO96 Register (Offset = 20Ch) [reset = 0h]

UID119TO96 is shown in Figure 4-802 and described in Table 4-839.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-218. UID119TO96 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID119TO96																															
R-0h																															

Table 4-236. UID119TO96 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID119TO96	R	0h	FROM1 Efuse Read UID[119:96]

4.7.1.37 MEMINITSTARTSHMEM Register (Offset = 2A8h) [reset = 0h]

MEMINITSTARTSHMEM is shown in Figure 4-803 and described in Table 4-840.

Return to Summary Table.

Shared memory initialization start

Figure 4-219. MEMINITSTARTSHMEM Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
MEMINITSTARTBANK7	MEMINITSTARTBANK6	MEMINITSTARTBANK5	MEMINITSTARTBANK4	MEMINITSTARTBANK3	MEMINITSTARTBANK2	MEMINITSTARTBANK1	MEMINITSTARTBANK0
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-237. MEMINITSTARTSHMEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1		0h	Not used
7	MEMINITSTARTBANK7		0h	Writing '1' will trigger Shared memory initialization for bank 7.Self clearing
6	MEMINITSTARTBANK6		0h	Writing '1' will trigger Shared memory initialization for bank 6.Self clearing
5	MEMINITSTARTBANK5		0h	Writing '1' will trigger Shared memory initialization for bank 5.Self clearing
4	MEMINITSTARTBANK4		0h	Writing '1' will trigger Shared memory initialization for bank 4.Self clearing
3	MEMINITSTARTBANK3		0h	Writing '1' will trigger Shared memory initialization for bank 3.Self clearing
2	MEMINITSTARTBANK2		0h	Writing '1' will trigger Shared memory initialization for bank 2.Self clearing
1	MEMINITSTARTBANK1		0h	Writing '1' will trigger Shared memory initialization for bank 1.Self clearing
0	MEMINITSTARTBANK0		0h	Writing '1' will trigger Shared memory initialization for bank 0.Self clearing

4.7.1.38 MEMINITDONESHMEM Register (Offset = 2ACh) [reset = 0h]

MEMINITDONESHMEM is shown in Figure 4-804 and described in Table 4-841.

Return to Summary Table.

Shared memory intialization end

Figure 4-220. MEMINITDONESHMEM Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
MEMINITDONEBANK7	MEMINITDONEBANK6	MEMINITDONEBANK5	MEMINITDONEBANK4	MEMINITDONEBANK3	MEMINITDONEBANK2	MEMINITDONEBANK1	MEMINITDONEBANK0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 4-238. MEMINITDONESHMEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1		0h	Not used
7	MEMINITDONEBANK7	R	0h	Memory Initialization done status for Shared memory for bank 7
6	MEMINITDONEBANK6	R	0h	Memory Initialization done status for Shared memory for bank 6
5	MEMINITDONEBANK5	R	0h	Memory Initialization done status for Shared memory for bank 5
4	MEMINITDONEBANK4	R	0h	Memory Initialization done status for Shared memory for bank 4
3	MEMINITDONEBANK3	R	0h	Memory Initialization done status for Shared memory for bank 3
2	MEMINITDONEBANK2	R	0h	Memory Initialization done status for Shared memory for bank 2
1	MEMINITDONEBANK1	R	0h	Memory Initialization done status for Shared memory for bank 1
0	MEMINITDONEBANK0	R	0h	Memory Initialization done status for Shared memory for bank 0

4.7.1.39 DSSMEMTAB0 Register (Offset = 2B0h) [reset = 76543210h]

DSSMEMTAB0 is shown in Figure 4-805 and described in Table 4-842.

Return to Summary Table.

Controls ordering of banks in shared memory associated with DSS

Figure 4-221. DSSMEMTAB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSSMEMTAB0																															
R/W-76543210h																															

Table 4-239. DSSMEMTAB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DSSMEMTAB0	R/W	76543210h	DSS L3RAM memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 0), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 1), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 2), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 3). 4th 128KB address goes to bank number programmed in [19:16] of this register(default is bank 4), 5th 128KB address goes to bank number programmed in [23:20] of this register(default is bank 5), 6th 128KB address goes to bank number programmed in [27:24] of this register(default is bank 6), 7th 128KB address goes to bank number programmed in [31:28] of this register(default is bank 7). Corresponding banks need to be selected for DSS in SHMEMBANKSEL regsiter

4.7.1.40 TCMAMEMTAB Register (Offset = 2BCh) [reset = 7654h]

TCMAMEMTAB is shown in Figure 4-806 and described in Table 4-843.

Return to Summary Table.

Controls ordering of banks in shared memory associated with MSS TCMA

Figure 4-222. TCMAMEMTAB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																TCMAMEMTAB															
0h																R/W-7654h															

Table 4-240. TCMAMEMTAB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU1		0h	Not Used
15-0	TCMAMEMTAB	R/W	7654h	MSS TCMA memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 4), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 5), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 6), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 7). Corresponding banks need to be selected for MSS TCMA in SHMEMBANKSEL regsiter

4.7.1.41 TCMBMEMTAB Register (Offset = 2C0h) [reset = 7654h]

TCMBMEMTAB is shown in Figure 4-807 and described in Table 4-844.

Return to Summary Table.

Controls ordering of banks in shared memory associated with MSS TCMB

Figure 4-223. TCMBMEMTAB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																TCMBMEMTAB															
0h																R/W-7654h															

Table 4-241. TCMBMEMTAB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU1		0h	Not Used
15-0	TCMBMEMTAB	R/W	7654h	MSS TCMB memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 4), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 5), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 6), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 7). Corresponding banks need to be selected for MSS TCMB in SHMEMBANKSEL regsiter

4.7.1.42 SHMEMBANKSEL3TO0 Register (Offset = 2C8h) [reset = 01010101h]

SHMEMBANKSEL3TO0 is shown in Figure 4-808 and described in Table 4-845.

Return to Summary Table.

Shared memory master allocation.Writing to each 8 bit field indicates the bank allocated to which master. 0x1 : DSS 0x2 : MSS TCMA 0x4 : MSS TCMB 0x8 : OCLA 0x10: BSS TCMA

Figure 4-224. SHMEMBANKSEL3TO0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BANK3								BANK2								BANK1								BANK0							
R/W-1h								R/W-1h								R/W-1h								R/W-1h							

Table 4-242. SHMEMBANKSEL3TO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BANK3	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
23-16	BANK2	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
15-8	BANK1	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
7-0	BANK0	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)

4.7.1.43 SHMEMBANKSEL7TO4 Register (Offset = 2CCh) [reset = 01010101h]

SHMEMBANKSEL7TO4 is shown in Figure 4-809 and described in Table 4-846.

Return to Summary Table.

Shared memory master allocation.Writing to each 8 bit field indicates the bank allocated to which master. 0x1 : DSS 0x2 : MSS TCMA 0x4 : MSS TCMB 0x8 : OCLA 0x10: BSS TCMA

Figure 4-225. SHMEMBANKSEL7TO4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BANK7								BANK6								BANK5								BANK4							
R/W-1h								R/W-1h								R/W-1h								R/W-1h							

Table 4-243. SHMEMBANKSEL7TO4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BANK7	R/W	1h	only valid value is 0x1/0x2/0x4/0x8/0X10 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA,BSS TCMA)
23-16	BANK6	R/W	1h	only valid value is 0x1/0x2/0x4/0x8/0X10 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA,BSS TCMA)
15-8	BANK5	R/W	1h	only valid value is 0x1/0x2/0x4/0x8 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA)
7-0	BANK4	R/W	1h	only valid value is 0x1/0x2/0x4/0x8 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA)

4.7.1.44 PBISTCLKCTL Register (Offset = 2D0h) [reset = 1h]

PBISTCLKCTL is shown in Figure 4-810 and described in Table 4-847.

Return to Summary Table.

PBIST clock control register

Figure 4-226. PBISTCLKCTL Register

31	30	29	28	27	26	25	24
NU							
0h							
23	22	21	20	19	18	17	16
NU							
0h							
15	14	13	12	11	10	9	8
PBIST300MCLKGATE							
R/W-0h							
7	6	5	4	3	2	1	0
PBIST300MCLKSRCSEL				PBIST300MCLKDIV			
R/W-0h				R/W-1h			

Table 4-244. PBISTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Not used
15-8	PBIST300MCLKGATE	R/W	0h	Pre clock divider gate for PBIST300M clock. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA
7-4	PBIST300MCLKSRCSEL	R/W	0h	Select clock source for DSP PBIST Clock source 000 => CPUCLK 001, 100, 101, 111 => RCCLK 010 => 600-MHz PLL divided clock 110 => REFCLK from ANA
3-0	PBIST300MCLKDIV	R/W	1h	Divide value for DSP PBIST source clock selected by field PBIST300MCLKSRCSEL in register CLKSRCSEL. “0000” => div1 “0001” => div2 | | “1111” => div15 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming PBIST300MCLKSRCSEL

4.7.2 MSS_RCM Registers

Table 4-848 lists the memory-mapped registers for the MSS_RCM. All register offset addresses not listed in Table 4-848 should be considered as reserved locations and the register contents should not be modified.

Table 4-245. MSS_RCM Registers

Offset	Acronym	Register Name	Section
4h	SOFTRST1	SYS_SOFT_RESET1	Section 4.9.2.1
8h	SOFTRST2	SYS_SOFT_RESET2	Section 4.9.2.2
18h	CLKDIVCTL0	CLKDIV	Section 4.9.2.3
1Ch	CLKSRCSEL0	CLKSRCSEL	Section 4.9.2.4
20h	CR4CTL	CR4CTL	Section 4.9.2.5
3Ch	CLKGATE	CLK_GATE	Section 4.9.2.6
44h	CLKSRCSEL1	SYS_CLKSRCSEL	Section 4.9.2.7
54h	CURRCLKDIV0	CURR_CLKDIV	Section 4.9.2.8
58h	RTICURRCLKDIV	RTI_CURR_CLKDIV	Section 4.9.2.9
5Ch	MEMINITSTART	MEM_INIT_START	Section 4.9.2.10
60h	CURRCLKDIV1	CURR_CLKDIV1	Section 4.9.2.11
6Ch	MEMINITDONE	MEM_INIT_DONE	Section 4.9.2.12
70h	ECCENMSSGEM	ECCEN_MSSGEM	Section 4.9.2.13
74h	ECCCAPTMSSGEM	ECCCAPT_MSSGEM	Section 4.9.2.14
78h	ECCENBSSGEM	ECCEN_BSSGEM	Section 4.9.2.15
7Ch	ECCCAPTBSSGEM	ECCCAPT_BSSGEM	Section 4.9.2.16
80h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.3.19
84h	NSYSPERUSERMODEN	NON_SYS_PERIPH_USERMODEEN	Section 4.9.2.18
88h	SECURERAMMMI	SECURERAMMMI	Section 4.9.2.19
8Ch	SECURERAMECC	SECURERAMECC	Section 4.9.2.20
90h	ESMGATE0	ESMGATE0	Section 4.9.2.21
94h	ESMGATE1	ESMGATE1	Section 4.9.2.22
98h	ESMGATE2	ESMGATE2	Section 4.9.2.23
9Ch	ESMGATE3	ESMGATE3	Section 4.9.2.24
A0h	ESMGATE4	ESMGATE4	Section 4.9.2.25
ACh	KEY	CFGREG_ACCESS_KEY	Section 4.9.2.26
B8h	SWIRQA	SWIRQ0	Section 4.9.2.27
BCh	SWIRQB	SWIRQ1	Section 4.9.2.28
C0h	MISCCTL0	MISCELLANEOUS_CTL_REG	Section 4.9.2.29
C4h	ATCMERRCAPTCTL	ATCMERRCAPT	Section 4.9.2.30
C8h	B0TCMERRCAPTCTL	B0TCMERRCAPT	Section 4.9.2.31
CCh	B1TCMERRCAPTCTL	B1TCMERRCAPT	Section 4.9.2.32
D0h	SOFTCORERST	SOFT_CORE_RST	Section 4.9.2.33
D8h	RSTCAUSE	MSS_RST_CAUSE	Section 4.9.2.34
DCh	RSTCAUSECLR	MSS_RST_CAUSE_CLR	Section 4.9.2.35
E0h	SPITRIGSRC	SPI_TRIG_SRC	Section 4.9.2.36
E4h	CLKINUSE	MSS_CLK_IN_USE	Section 4.9.2.37
E8h	ECCENMSSBSS	MSS_ECC_EN	Section 4.9.2.38
ECh	ECCCAPTMSSBSS	MSS_ECC_CAPT	Section 4.9.2.39
F0h	CLKDIVCTL2	CLKDIV2	Section 4.9.2.40
FCh	SWIRQC	SW_IRQ2	Section 4.9.2.41

Complex bit access types are encoded to fit into small table cells. Table 4-849 shows the codes that are used for access types in this section.

Table 4-246. MSS_RCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.7.2.1 SOFTRST1 Register (Offset = 4h) [reset = 0h]

SOFTRST1 is shown in Figure 4-811 and described in Table 4-850.

Return to Summary Table.

Figure 4-227. SOFTRST1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CR4SYSRST							
R/W-0h								0h							

Table 4-247. SOFTRST1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	CR4SYSRST		0h	Write 0xAD to assert a MSS CR4 only reset.Self clearing By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

NOTE: This register used for the ROM eclipsing feature. Refer to the ROM Eclipsing section for more details.

4.7.2.2 SOFTRST2 Register (Offset = 8h) [reset = 0h]

SOFTRST2 is shown in Figure 4-812 and described in Table 4-851.

Return to Summary Table.

Figure 4-228. SOFTRST2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIMRST								RESERVED																							
R/W-0h								R/W-0h																							

Table 4-248. SOFTRST2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VIMRST	R/W	0h	Write 0xAD to assert a VIM only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-0	RESERVED	R/W	0h	Reserved

4.7.2.3 CLKDIVCTL0 Register (Offset = 18h) [reset = 0h]

CLKDIVCTL0 is shown in Figure 4-813 and described in Table 4-852.

Return to Summary Table.

Figure 4-229. CLKDIVCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCANCLKDIV								RESERVED							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCLKDIV								RESERVED							
R/W-0h								R/W-0h							

Table 4-249. CLKDIVCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FDCANCLKDIV	R/W	0h	Divide value for FDCAN source clock selected by field FDCANCLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
23-16	RESERVED	R/W	0h	Reserved
15-8	VCLKCLKDIV	R/W	0h	Divide value for MSS subsystem source clock selected by field VCLKCLKSRCSEL in register CLKSRCSEL1 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
7-0	RESERVED	R/W	0h	Reserved

4.7.2.4 CLKSRCSEL0 Register (Offset = 1Ch) [reset = 0h]

CLKSRCSEL0 is shown in Figure 4-814 and described in Table 4-853.

Return to Summary Table.

Figure 4-230. CLKSRCSEL0 Register

31	30	29	28	27	26	25	24
RTIDCLKSRCSEL				RTICCLKSRCSEL			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
RESERVED				QSPICLKSRCSEL			
0h				R/W-0h			
15	14	13	12	11	10	9	8
RESERVED				FDCANCLKSRCSEL			
0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED							
0h							

Table 4-250. CLKSRCSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RTIDCLKSRCSEL	R/W	0h	Select clock source for RTID baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
27-24	RTICCLKSRCSEL	R/W	0h	Select clock source for RTIC baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
23-20	RESERVED		0h	
19-16	QSPICLKSRCSEL	R/W	0h	Select clock source for QSPI baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
15-12	RESERVED		0h	
11-8	FDCANCLKSRCSEL	R/W	0h	Select clock source for FDCANCLKSRCSEL 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA 111 => RCCLK
7-0	RESERVED		0h	

4.7.2.5 CR4CTL Register (Offset = 20h) [reset = 0h]

CR4CTL is shown in Figure 4-815 and described in Table 4-854.

Return to Summary Table.

Figure 4-231. CR4CTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU								MEMSWAPWAIT							
0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR4MEMSWAP								RESERVED							
R/W-0h								R/W-0h							

Table 4-251. CR4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU		0h	Reserved
23-16	MEMSWAPWAIT	R/W	0h	When CR4MEMSWAP is 0xAD : Write 0xAD to this field enable the CR4MEMSWAP only after a CR4 reset – either by writing to CR4SYSRST or by writing to PRCR register in CR4 debug space.
15-8	CR4MEMSWAP	R/W	0h	Write 0xAD will map the MSS CR4 0x0000_0000 to MSS CR4 TCMA RAM start address.
7-0	RESERVED	R/W	0h	Reserved

4.7.2.6 CLKGATE Register (Offset = 3Ch) [reset = 400h]

CLKGATE is shown in Figure 4-816 and described in Table 4-855.

Return to Summary Table.

Figure 4-232. CLKGATE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RTIDCLKGATE	RTICCLKGATE	FDCANCLKGATE	RESERVED	QSPICLKGATE	RESERVED		
R/W-0h	R/W-0h	R/W-0h	0h	R/W-0h	0h		

Table 4-252. CLKGATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED		0h	Reserved
7	RTIDCLKGATE	R/W	0h	Pre clock divider gate for RTID clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
6	RTICCLKGATE	R/W	0h	Pre clock divider gate for RTIC clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
5	FDCANCLKGATE	R/W	0h	Pre clock divider gate for FDCAN clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
4	RESERVED	R/W	0h	Reserved
3	QSPICLKGATE	R/W	0h	Pre clock divider gate for QSPI clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
2-0	RESERVED	R/W	0h	Reserved

4.7.2.7 CLKSRCSEL1 Register (Offset = 44h) [reset = 0h]

CLKSRCSEL1 is shown in Figure 4-817 and described in Table 4-856.

Return to Summary Table.

Figure 4-233. CLKSRCSEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RESERVED				VCLKCLKSRCSEL			
0h				R/W-0h			

Table 4-253. CLKSRCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED		0h	Reserved
3-0	VCLKCLKSRCSEL	R/W	0h	Select clock source for MSS subsystem VCLK 000 => CPUCLK 00, 100, 101, 111 => RCCLK 010 => 600-MHz PLL divided clock 100 => CPUCLK 110 => REFCLK from ANA

4.7.2.8 CURRCLKDIV0 Register (Offset = 54h) [reset = 0h]

CURRCLKDIV0 is shown in Figure 4-818 and described in Table 4-857.

Return to Summary Table.

Figure 4-234. CURRCLKDIV0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCANCURRCLKDIV								RESERVED							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCURRCLKDIV								RESERVED							
R-0h								0h							

Table 4-254. CURRCLKDIV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FDCANCURRCLKDIV	R	0h	Returns Current divide value of FDCAN baud clock divider.
23-16	RESERVED	R	0h	Reserved
15-8	VCLKCURRCLKDIV	R	0h	Returns Current divide value of VCLK divider.
7-0	RESERVED		0h	Reserved

4.7.2.9 RTICURRCLKDIV Register (Offset = 58h) [reset = 0h]

RTICURRCLKDIV is shown in Figure 4-819 and described in Table 4-858.

Return to Summary Table.

Figure 4-235. RTICURRCLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU								NU							
0h								0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTCURRCLKDIV								RTICURRCLKDIV							
R-0h								R-0h							

Table 4-255. RTICURRCLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	
15-8	WDTCURRCLKDIV	R	0h	Not used.Reserved
7-0	RTICURRCLKDIV	R	0h	Not used.Reserved

4.7.2.10 MEMINITSTART Register (Offset = 5Ch) [reset = 0h]

MEMINITSTART is shown in Figure 4-820 and described in Table 4-859.

Return to Summary Table.

Figure 4-236. MEMINITSTART Register

31	30	29	28	27	26	25	24
MEMINITKEY							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED						BSSMBOX4GEMMEM	MSSMBOX4GEMMEM
0h						0h	0h
15	14	13	12	11	10	9	8
GEMMBOX4MSSMEM	GEMMBOX4BSSMEM	RESERVED				DMA2MEM	BSSMBOX4MSSMEM
0h	0h	0h				0h	0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	RESERVED	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-256. MEMINITSTART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MEMINITKEY	R/W	0h	Memory hardware initialization global enable key. Write 0XAD to enable MEMINIT.
23-18	RESERVED		0h	Reserved
17	BSSMBOX4GEMMEM		0h	Writing '1' will trigger DSS- BSS mailbox initialization.Self clearing
16	MSSMBOX4GEMMEM		0h	Writing '1' will trigger DSS- MSS mailbox initialization.Self clearing
15	GEMMBOX4MSSMEM		0h	Writing '1' will trigger DSS- MSS mailbox initialization.Self clearing
14	GEMMBOX4BSSMEM		0h	Writing '1' will trigger DSS- BSS mailbox initialization.Self clearing
13-10	RESERVED		0h	Reserved
9	DMA2MEM		0h	Writing '1' will trigger DMA2 memory initialization.Self clearing
8	BSSMBOX4MSSMEM		0h	Writing '1' will trigger MSS- BSS mailbox initialization.Self clearing
7	MSSMBOX4BSSMEM		0h	Writing '1' will trigger MSS- BSS mailbox initialization.Self clearing
6	RESERVED		0h	Reserved
5	SPIBMEM		0h	Writing '1' will trigger SPIB memory initialization.Self clearing
4	SPIAMEM		0h	Writing '1' will trigger SPIA memory initialization.Self clearing
3	VIMMEM		0h	Writing '1' will trigger VIM memory initialization.Self clearing
2	DMAMEM		0h	Writing '1' will trigger DMA memory initialization.Self clearing
1	CR4TCMBMEM		0h	Writing '1' will trigger MSS TCMB memory initialization.Self clearing
0	CR4TCMAMEM		0h	Writing '1' will trigger MSS TCMA memory initialization.Self clearing

4.7.2.11 CURRCLKDIV1 Register (Offset = 60h) [reset = 0h]

CURRCLKDIV1 is shown in Figure 4-821 and described in Table 4-860.

Return to Summary Table.

Figure 4-237. CURRCLKDIV1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								QSPICURRCLKDIV							
R-0h								R-0h							

Table 4-257. CURRCLKDIV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	QSPICURRCLKDIV	R	0h	Returns Current divide value of QSPI_CLK divider.

4.7.2.12 MEMINITDONE Register (Offset = 6Ch) [reset = 0h]

MEMINITDONE is shown in Figure 4-822 and described in Table 4-861.

Return to Summary Table.

Figure 4-238. MEMINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED						BSSMBOX4GEMMEM	MSSMBOX4GEMMEM
0h						R-0h	R-0h
15	14	13	12	11	10	9	8
GEMMBOX4MSSMEM	GEMMBOX4BSSMEM	RESERVED				DMA2MEM	BSSMBOX4MSSMEM
R-0h	R-0h	R-0h				R-0h	R-0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	RESERVED	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 4-258. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED		0h	Reserved
17	BSSMBOX4GEMMEM	R	0h	Memory Initialization done status for DSS- BSS mailbox
16	MSSMBOX4GEMMEM	R	0h	Memory Initialization done status for DSS- MSS mailbox
15	GEMMBOX4MSSMEM	R	0h	Memory Initialization done status for DSS- MSS mailbox
14	GEMMBOX4BSSMEM	R	0h	Memory Initialization done status for DSS- BSS mailbox
13-10	RESERVED	R	0h	Reserved
9	DMA2MEM	R	0h	Memory Initialization done status for MSS DMA2 memory
8	BSSMBOX4MSSMEM	R	0h	Memory Initialization done status for MSS- BSS mailbox
7	MSSMBOX4BSSMEM	R	0h	Memory Initialization done status for MSS- BSS mailbox
6	RESERVED	R	0h	Reserved
5	SPIBMEM	R	0h	Memory Initialization done status for MSS SPIB memory
4	SPIAMEM	R	0h	Memory Initialization done status for MSS SPIA memory
3	VIMMEM	R	0h	Memory Initialization done status for MSS VIM memory
2	DMAMEM	R	0h	Memory Initialization done status for MSS DMA memory
1	CR4TCMBMEM	R	0h	Memory Initialization done status for MSS TCMB memory
0	CR4TCMAMEM	R	0h	Memory Initialization done status for MSS TCMA memory

4.7.2.13 ECCENMSSGEM Register (Offset = 70h) [reset = 0h]

ECCENMSSGEM is shown in Figure 4-823 and described in Table 4-862.

Return to Summary Table.

Figure 4-239. ECCENMSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENMSSGEM																															
R/W-0h																															

Table 4-259. ECCENMSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENMSSGEM	R/W	0h	7:0 : Writing 0xAD will enable ECC for MSS- DSS mailbox 15:8: Writing 0xAD will enable ECC for MSS- DSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in MSS mailbox for GEM 21:19: Write 3'b111 to clear the Address captured because of ECC error. In GEM mailbox for MSS

4.7.2.14 ECCCAPTMSSGEM Register (Offset = 74h) [reset = 0h]

ECCCAPTMSSGEM is shown in Figure 4-824 and described in Table 4-863.

Return to Summary Table.

Figure 4-240. ECCCAPTMSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTMSSGEM																															
R-0h																															

Table 4-260. ECCCAPTMSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTMSSGEM	R	0h	7:0 : mss_mbox4gem_ecc_fault_address 14:8 : mss_mbox4gem_repaired_bit 23:16 : gem_mbox4mss_ecc_fault_address 30:24 : gem_mbox4mss_repaired_bit

4.7.2.15 ECCENBSSGEM Register (Offset = 78h) [reset = 0h]

ECCENBSSGEM is shown in Figure 4-825 and described in Table 4-864.

Return to Summary Table.

Figure 4-241. ECCENBSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENBSSGEM																															
R/W-0h																															

Table 4-261. ECCENBSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENBSSGEM	R/W	0h	7:0 : Writing 0xAD will enable ECC for DSS- BSS mailbox 15:8: Writing 0xAD will enable ECC for DSS- BSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in GEM mailbox for BSS 21:19: Write 3'b111 to clear the Address captured because of ECC error in BSS mailbox for GEM

4.7.2.16 ECCCAPTBSSGEM Register (Offset = 7Ch) [reset = 0h]

ECCCAPTBSSGEM is shown in Figure 4-826 and described in Table 4-865.

Return to Summary Table.

Figure 4-242. ECCCAPTBSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTBSSGEM																															
R-0h																															

Table 4-262. ECCCAPTBSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTBSSGEM	R	0h	7:0 : bss_mbox4gem_ecc_fault_address 14:8 : bss_mbox4gem_repaired_bit 23:16 : gem_mbox4bss_ecc_fault_address 30:24 : gem_mbox4bss_repaired_bit

4.7.2.17 USERMODEEN Register (Offset = 80h) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-243. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-263. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS RCM space.

4.7.2.18 NSYSPERUSERMODEN Register (Offset = 84h) [reset = 0h]

NSYSPERUSERMODEN is shown in Figure 4-828 and described in Table 4-867.

Return to Summary Table.

Figure 4-244. NSYSPERUSERMODEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSYSPERUSERMODEN																															
R/W-0h																															

Table 4-264. NSYSPERUSERMODEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NSYSPERUSERMODEN	R/W	0h	2:0 : Write 3'b111 to enable user mode access to SPIA 5:3 : Write 3'b111 to enable user mode access to SPIB 10:8 : Write 3'b111 to enable user mode access to GIO 13:11 : Write 3'b111 to enable user mode access to QSPI 18:16 : Write 3'b111 to enable user mode access to SCIA 21:19 : Write 3'b111 to enable user mode access to SCIB 26:24

4.7.2.19 SECURERAMMMI Register (Offset = 88h) [reset = 0h]

SECURERAMMMI is shown in Figure 4-829 and described in Table 4-868.

Return to Summary Table.

Figure 4-245. SECURERAMMMI Register

31	30	29	28	27	26	25	24
NU2							
0h							
23	22	21	20	19	18	17	16
NU2							SECURERAMINITDONE
0h							R-0h
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
NU1							SECURERAMINIT
0h							0h

Table 4-265. SECURERAMMMI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU2		0h	
16	SECURERAMINITDONE	R	0h	Memory Initialization done status for Secure Key RAM
15-1	NU1		0h	
0	SECURERAMINIT		0h	Writing '1' will trigger Secure Key RAM memory initialization.Self clearing

4.7.2.20 SECURERAMECC Register (Offset = 8Ch) [reset = 0h]

SECURERAMECC is shown in Figure 4-830 and described in Table 4-869.

Return to Summary Table.

Figure 4-246. SECURERAMECC Register

31	30	29	28	27	26	25	24
SECURERAMBIT							
R-0h							
23	22	21	20	19	18	17	16
SECURERAMADDR							
R-0h							
15	14	13	12	11	10	9	8
NU					SECURERAMECCCLR		
0h					R/W-0h		
7	6	5	4	3	2	1	0
SECURERAMECCEN							
R/W-0h							

Table 4-266. SECURERAMECC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SECURERAMBIT	R	0h	Secure key RAM repaired bit
23-16	SECURERAMADDR	R	0h	Secure Key RAM_ecc_fault_address
15-11	NU		0h	
10-8	SECURERAMECCCLR	R/W	0h	10:8 : Write 3'b111 to clear the Address captured because of ECC error
7-0	SECURERAMECCEN	R/W	0h	7:0 : Writing 0xAD will enable ECC for Secure key RAM

4.7.2.21 ESMGATE0 Register (Offset = 90h) [reset = 0h]

ESMGATE0 is shown in Figure 4-831 and described in Table 4-870.

Return to Summary Table.

Figure 4-247. ESMGATE0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE0																															
R/W-07070700h																															

Table 4-267. ESMGATE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE0	R/W	07070700h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 0 [7:4] : Gate ESM group2 line 1 [11:8] : Gate ESM group2 line 2 [15:12] : Gate ESM group2 line 3 [19:16] : Gate ESM group2 line 4 [23:20] : Gate ESM group2 line 5 [27:24] : Gate ESM group2 line 6 [31:28] : Gate ESM group2 line 7 Static register setting.Should not be changed on the fly

4.7.2.22 ESMGATE1 Register (Offset = 94h) [reset = 0h]

ESMGATE1 is shown in Figure 4-832 and described in Table 4-871.

Return to Summary Table.

Figure 4-248. ESMGATE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE1																															
R/W-7h																															

Table 4-268. ESMGATE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE1	R/W	7h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 8 [7:4] : Gate ESM group2 line 9 [11:8] : Gate ESM group2 line 10 [15:12] : Gate ESM group2 line 11 [19:16] : Gate ESM group2 line 12 [23:20] : Gate ESM group2 line 13 [27:24] : Gate ESM group2 line 14 [31:28] : Gate ESM group2 line 15 Static register setting.Should not be changed on the fly

4.7.2.23 ESMGATE2 Register (Offset = 98h) [reset = 0h]

ESMGATE2 is shown in Figure 4-833 and described in Table 4-872.

Return to Summary Table.

Figure 4-249. ESMGATE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE2																															
R/W-7h																															

Table 4-269. ESMGATE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE2	R/W	7h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23 Static register setting should not be changed on the fly

4.7.2.24 ESMGATE3 Register (Offset = 9Ch) [reset = 0h]

ESMGATE3 is shown in Figure 4-834 and described in Table 4-873.

Return to Summary Table.

Figure 4-250. ESMGATE3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE3																															
R/W-3Fh																															

Table 4-270. ESMGATE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE3	R/W	3Fh	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23 Static register setting.Should not be changed on the fly

4.7.2.25 ESMGATE4 Register (Offset = A0h) [reset = 0h]

ESMGATE4 is shown in Figure 4-835 and described in Table 4-874.

Return to Summary Table.

Figure 4-251. ESMGATE4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE4																															
R/W-0h																															

Table 4-271. ESMGATE4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE4	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group3 line 0 [7:4] : Gate ESM group3 line 1 [11:8] : Gate ESM group3 line 2 [15:12] : Gate ESM group3 line 3 [19:16] : Gate ESM group3 line 4 [23:20] : Gate ESM group3 line 5 [27:24] : Gate ESM group3 line 6 [31:28] : Gate ESM group3 line 7 Static register setting.Should not be changed on the fly

4.7.2.26 KEY Register (Offset = ACh) [reset = 83E783E7h]

KEY is shown in Figure 4-836 and described in Table 4-875.

Return to Summary Table.

Figure 4-252. KEY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY																															
R/W-83E783E7h																															

Table 4-272. KEY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY	R/W	83E783E7h	Kicker Register. The value 83E7_83E7h must be written as part of the process to unlock the CPU write access to the MSS RCM registers

4.7.2.27 SWIRQA Register (Offset = B8h) [reset = 0h]

SWIRQA is shown in Figure 4-837 and described in Table 4-876.

Return to Summary Table.

Figure 4-253. SWIRQA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ1								SWIRQ1DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ0								SWIRQ0DAT							
R/W-0h								R/W-0h							

Table 4-273. SWIRQA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ1	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ1DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ0	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ0DAT	R/W	0h	Not Used.Reserved

4.7.2.28 SWIRQB Register (Offset = BCh) [reset = 0h]

SWIRQB is shown in Figure 4-838 and described in Table 4-877.

Return to Summary Table.

Figure 4-254. SWIRQB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ3								SWIRQ3DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ2								SWIRQ2DAT							
R/W-0h								R/W-0h							

Table 4-274. SWIRQB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ3	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ3DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ2	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ2DAT	R/W	0h	Not Used.Reserved

4.7.2.29 MISCCTL0 Register (Offset = C0h) [reset = ADADh]

MISCCTL0 is shown in Figure 4-839 and described in Table 4-878.

Return to Summary Table.

Figure 4-255. MISCCTL0 Register

31	30	29	28	27	26	25	24
NU							TCMB1EZDIS
0h							R/W-0h
23	22	21	20	19	18	17	16
TCMB1EZDIS			TCMB0EZDIS			TCMAEZDIS	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							
R/W-ADh							
7	6	5	4	3	2	1	0
RESERVED							
R/W-ADh							

Table 4-275. MISCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU		0h	
24-22	TCMB1EZDIS	R/W	0h	Write 111 to make TCMB1 EZ to '1' regardless of Functional value.
21-19	TCMB0EZDIS	R/W	0h	Write 111 to make TCMB0 EZ to '1' regardless of Functional value.
18-16	TCMAEZDIS	R/W	0h	Write 111 to make TCMA EZ to '1' regardless of Functional value.
15-0	RESERVED	R/W	ADh	Reserved

4.7.2.30 ATCMERRCAPTCTL Register (Offset = C4h) [reset = 0h]

ATCMERRCAPTCTL is shown in Figure 4-840 and described in Table 4-879.

Return to Summary Table.

Figure 4-256. ATCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRATCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRATCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRATCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		ATCFORCEERR			ERRATCADDRCLR		
0h		R/W-0h			0h		

Table 4-276. ATCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRATCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	ATCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRATCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.7.2.31 B0TCMERRCAPTCTL Register (Offset = C8h) [reset = 0h]

B0TCMERRCAPTCTL is shown in Figure 4-841 and described in Table 4-880.

Return to Summary Table.

Figure 4-257. B0TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB0TCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRB0TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB0TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B0TCFORCEERR			ERRB0TCADDRCLR		
0h		R/W-0h			0h		

Table 4-277. B0TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRB0TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	B0TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB0TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.7.2.32 B1TCMERRCAPTCTL Register (Offset = CCh) [reset = 0h]

B1TCMERRCAPTCTL is shown in Figure 4-842 and described in Table 4-881.

Return to Summary Table.

Figure 4-258. B1TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB1TCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRB1TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB1TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B1TCFORCEERR			ERRB1TCADDRCLR		
0h		R/W-0h			0h		

Table 4-278. B1TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRB1TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	B1TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB1TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.7.2.33 SOFTCORERST Register (Offset = D0h) [reset = 000F0F00h]

SOFTCORERST is shown in Figure 4-843 and described in Table 4-882.

Return to Summary Table.

Figure 4-259. SOFTCORERST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RST_WFICHECKEN								RESERVED							
R/W-0h								R/W-Fh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSTTOASSRTDLY								RESERVED							
R/W-Fh								0h							

Table 4-279. SOFTCORERST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RST_WFICHECKEN	R/W	0h	0xAD : When CR4SYSRST is set, Before asserting the reset to CR4 wait for CR4 to enter WFI. 0X0: Dont wait for WFI
23-16	RESERVED	R/W	Fh	Reserved
15-8	RSTTOASSRTDLY	R/W	Fh	Wait for programmed number of clock cycle before reset is asserted to CR4.
7-0	RESERVED		0h	Reserved

4.7.2.34 RSTCAUSE Register (Offset = D8h) [reset = 0h]

RSTCAUSE is shown in Figure 4-844 and described in Table 4-883.

Return to Summary Table.

Figure 4-260. RSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								RSTCAUSE							
0h																								R-0h							

Table 4-280. RSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	RSTCAUSE	R	0h	MSS_RCM:RSTCAUSE 0000_1001 : System out of NRESET 0000_1000 : System out of Warm Reset 0010_0000 : CR4 reset because of Software trigger. 0001_0000 : CR4 STC reset 0100_0000 : CR4 reset because of writing to PRCR register in CR4 debug space. Note: Because the ROM Bootloader issues a soft reset to pass control to the application (or secondary bootloader), this register always reads 0x20. The reset value is stored by the ROM Bootloader into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.7.2.35 RSTCAUSECLR Register (Offset = DCh) [reset = 0h]

RSTCAUSECLR is shown in Figure 4-845 and described in Table 4-884.

Return to Summary Table.

Figure 4-261. RSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								RSTCAUSECLR							
0h								0h							

Table 4-281. RSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	
7-0	RSTCAUSECLR		0h	Write 0xAD to clear RSTCAUSE.Self clearing

4.7.2.36 SPITRIGSRC Register (Offset = E0h) [reset = 0h]

SPITRIGSRC is shown in Figure 4-846 and described in Table 4-885.

Return to Summary Table.

Figure 4-262. SPITRIGSRC Register

31	30	29	28	27	26	25	24
NU3					SPIBTRIG		
0h					R/W-0h		
23	22	21	20	19	18	17	16
SPIBTRIG							
R/W-0h							
15	14	13	12	11	10	9	8
NU2							SPIATRIG1
0h							R/W-0h
7	6	5	4	3	2	1	0
NU1							SPIATRIG0
0h							R/W-0h

Table 4-282. SPITRIGSRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU3		0h	
26-16	SPIBTRIG	R/W	0h	[20:16] --> Trigger sources for MIBSPIB
15-9	NU2		0h	
8	SPIATRIG1	R/W	0h	1st bit ofTRG_SRC input of SPIA.
7-1	NU1		0h	
0	SPIATRIG0	R/W	0h	0th bit ofTRG_SRC input of SPIA.

4.7.2.37 CLKINUSE Register (Offset = E4h) [reset = 0h]

CLKINUSE is shown in Figure 4-847 and described in Table 4-886.

Return to Summary Table.

Figure 4-263. CLKINUSE Register

31	30	29	28	27	26	25	24
RESERVED				FRCCLKINUSE			
0h				R-0h			
23	22	21	20	19	18	17	16
RTIDCLKINUSE				RTICCLKINUSE			
R-0h				R-0h			
15	14	13	12	11	10	9	8
QSPICLKINUSE				RESERVED			
R-0h				0h			
7	6	5	4	3	2	1	0
FDCANCLKINUSE				VCLKINUSE			
R-0h				R-0h			

Table 4-283. CLKINUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED		0h	Reserved
27-24	FRCCLKINUSE	R	0h	Current Clock Source Select Mux value for FRC CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
23-20	RTIDCLKINUSE	R	0h	Current Clock Source Select Mux value for RTID 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
19-16	RTICCLKINUSE	R	0h	Current Clock Source Select Mux value for RTIC 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
15-12	QSPICLKINUSE	R	0h	Current Clock Source Select Mux value for QSPI CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
11-8	RESERVED	R	0h	Reserved
7-4	FDCANCLKINUSE	R	0h	Current Clock Source Select Mux value for FDCAN CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
3-0	VCLKINUSE	R	0h	Current Clock Source Select Mux value for VCLK 000 => CPUCLK (40Mhz or 50 Mhz or 80Mh or 100Mh) 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 101 => RCCLK (10Mhz) 110 => REFCLK 111 => RCCLK (10Mhz)

4.7.2.38 ECCENMSSBSS Register (Offset = E8h) [reset = 0h]

ECCENMSSBSS is shown in Figure 4-848 and described in Table 4-887.

Return to Summary Table.

Figure 4-264. ECCENMSSBSS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENMSSBSS																															
R/W-0h																															

Table 4-284. ECCENMSSBSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENMSSBSS	R/W	0h	7:0 : Writing 0xAD will enable ECC for MSS – BSS mailbox 15:8: Writing 0xAD will enable ECC for MSS – BSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in MSS mailbox for BSS 21:19: Write 3'b111 to clear the Address captured because of ECC error. In BSS mailbox for MSS

4.7.2.39 ECCCAPTMSSBSS Register (Offset = ECh) [reset = 0h]

ECCCAPTMSSBSS is shown in Figure 4-849 and described in Table 4-888.

Return to Summary Table.

Figure 4-265. ECCCAPTMSSBSS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTMSSBSS																															
R-0h																															

Table 4-285. ECCCAPTMSSBSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTMSSBSS	R	0h	7:0 : mss_mbox4bss_ecc_fault_address 14:8 : mss_mbox4bss_repaired_bit 23:16 : bss_mbox4mss_ecc_fault_address 30:24 : bss_mbox4mss_repaired_bit 31 : Reserved

4.7.2.40 CLKDIVCTL2 Register (Offset = F0h) [reset = 0h]

CLKDIVCTL2 is shown in Figure 4-850 and described in Table 4-889.

Return to Summary Table.

Figure 4-266. CLKDIVCTL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								QSPICLKDIV							
0h								R/W-0h							

Table 4-286. CLKDIVCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	
7-0	QSPICLKDIV	R/W	0h	Divide value for QSPI baud clock selected by field QSPICLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256

4.7.2.41 SWIRQC Register (Offset = FCh) [reset = 0h]

SWIRQC is shown in Figure 4-851 and described in Table 4-890.

Return to Summary Table.

Figure 4-267. SWIRQC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ5								SWIRQ5DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ4								SWIRQ4DAT							
R/W-0h								R/W-0h							

Table 4-287. SWIRQC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ5	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ5DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ4	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ4DAT	R/W	0h	Not Used.Reserved

4.7.3 MSS_GPCFG_REG Registers

Table 4-891 lists the memory-mapped registers for the MSS_GPCFG_REG. All register offset addresses not listed in Table 4-891 should be considered as reserved locations and the register contents should not be modified.

Table 4-288. MSS_GPCFG_REG Registers

Offset	Acronym	Register Name	Section
0h	GPCFG0	GPCFG0	Section 4.9.3.1
4h	GPCFG1	GPCFG1	Section 4.9.3.2
8h	GPCFG2	GPCFG2	Section 4.9.3.3
Ch	GPCFG3	GPCFG3	Section 4.9.3.4
10h	GPCFG4	GPCFG4	Section 4.9.3.5
2Ch	GPCFG11	GPCFG11	Section 4.9.3.7
D0h	CCCACFG0	CCCA_CFG0	Section 4.9.3.8
D4h	CCCACFG1	CCCA_CFG1	Section 4.9.3.9
D8h	CCCACFG2	CCCA_CFG2	Section 4.9.3.10
DCh	CCCACFG3	CCCA_CFG3	Section 4.9.3.11
E0h	CCCBCFG0	CCCB_CFG0	Section 4.9.3.12
E4h	CCCBCFG1	CCCB_CFG1	Section 4.9.3.13
E8h	CCCBCFG2	CCCB_CFG2	Section 4.9.3.14
ECh	CCCBCFG3	CCCB_CFG3	Section 4.9.3.15
F0h	CCCACNTVAL	CCCACNTVAL	Section 4.9.3.16
F4h	CCCBCNTVAL	CCCBCNTVAL	Section 4.9.3.17
F8h	CCCABERRSTAT	CCCABERRSTAT	Section 4.9.3.18
FCh	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.3.19
140h	EPWMCFG	EPWMCFG	Section 4.9.3.20
148h	DMMSWINT0	DMMSWINT0	Section 4.9.3.21
14Ch	DMMSWINT1	DMMSWINT1	Section 4.9.3.22
150h	DMMSWINTSEL0	DMMSWINTSEL0	Section 4.9.3.23
154h	DMMSWINTSEL1	DMMSWINTSEL1	Section 4.9.3.24
158h	CCCBWDEN	CCCBWDEN	Section 4.9.3.25
15Ch	GPIOINTREDGESEL	GPIOINTREDGESEL	Section 4.9.3.26
164h	PWMDMATRIGEN	PWMDMATRIGEN	Section 4.9.3.27
168h	JTAGTXDATA	JTAGTXDATA	Section 4.9.3.28
16Ch	JTAGTXCONTROL	JTAGTXCONTROL	Section 4.9.3.29
170h	JTAGRXDATA	JTAGRXDATA	Section 4.9.3.30
178h	JTAGTXRXACK	JTAGTXRXACK	Section 4.9.3.31
17Ch	JTAGRXCONTROL	JTAGRXCONTROL	Section 4.9.3.32
180h	MSS2GEMSWIRQ		Section 4.9.3.33
184h	CSETBFLUSH		Section 4.9.3.34

Complex bit access types are encoded to fit into small table cells. Table 4-892 shows the codes that are used for access types in this section.

Table 4-289. MSS_GPCFG_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write

Table 4-289. MSS_GPCFG_REG Access Type Codes (continued)

Access Type	Code	Description
Reset or Default Value		
-n		Value after reset or the default value

4.7.3.1 GPCFG0 Register (Offset = 0h) [reset = 0h]

GPCFG0 is shown in Figure 4-852 and described in Table 4-893.

Return to Summary Table.

Figure 4-268. GPCFG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG0																															
R/W-0h																															

Table 4-290. GPCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG0	R/W	0h	General Purpose config register for SW use.

4.7.3.2 GPCFG1 Register (Offset = 4h) [reset = 0h]

GPCFG1 is shown in Figure 4-853 and described in Table 4-894.

Return to Summary Table.

Figure 4-269. GPCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG1																															
R/W-0h																															

Table 4-291. GPCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG1	R/W	0h	General Purpose config register for SW use.

4.7.3.3 GPCFG2 Register (Offset = 8h) [reset = 0h]

GPCFG2 is shown in Figure 4-854 and described in Table 4-895.

Return to Summary Table.

Figure 4-270. GPCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG2																															
R/W-0h																															

Table 4-292. GPCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG2	R/W	0h	General Purpose config register for SW use.

4.7.3.4 GPCFG3 Register (Offset = Ch) [reset = 0h]

GPCFG3 is shown in Figure 4-855 and described in Table 4-896.

Return to Summary Table.

Figure 4-271. GPCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG3																															
R/W-0h																															

Table 4-293. GPCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG3	R/W	0h	General Purpose config register for SW use.

4.7.3.5 GPCFG4 Register (Offset = 10h) [reset = 0h]

GPCFG4 is shown in Figure 4-856 and described in Table 4-897.

Return to Summary Table.

Figure 4-272. GPCFG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG4																															
R/W-0h																															

Table 4-294. GPCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG4	R/W	0h	General Purpose config register for SW use.

4.7.3.6 GPCFG6 Register (Offset = 18h) [reset = 0h]

GPCFG6 is shown in Figure 4-857 and described in Table 4-898.

Return to Summary Table.

Figure 4-273. GPCFG6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG6																															
R/W-0h																															

Table 4-295. GPCFG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG6	R/W	0h	Describes the muxing of Interrupts and DMA based on use cases - GPCFG6[0] = 1 : dma_req[36] = can_fd_intr[0] ,0 : GPCFG6[0] = 0 : dma_req[36] = can_fd_fe_intr[0] GPCFG6[1] = 1 : dma_req[44] = can_fd_intr[1] ,0 : GPCFG6[1] = 0 : dma_req[44] = can_fd_fe_intr[1] GPCFG6[2] = 1 : frc_can_intr[3] = can_fd_fe_intr[0] ,0 : GPCFG6[2] = 0 : frc_can_intr[3] = can_fd_intr[0] GPCFG6[3] = 1 : frc_can_intr[2] = can_fd_fe_intr[1] ,0 : GPCFG6[3] = 0 : frc_can_intr[2] = can_fd_intr[1] GPCFG6[8] = 1 : dma_req[5] = can_fd_fe_intr[2] ,0 : GPCFG6[8] = 0 : dma_req[5] = spia_dma_req[3] GPCFG6[11] = 1 : dma_req[37] = rti2 dma req[0] ,0 : GPCFG6[11] = 0 : dma_req[37] = spib_dma_req[2] GPCFG6[12] = 1 : dma_req[38] = rti2 dma req[1] ,0 : GPCFG6[12] = 0 : dma_req[38] = else spib_dma_req[3] GPCFG6[13] = 1 : dma_req[42] = rti3 dma req[0] ,0 : GPCFG6[13] = 0 : dma_req[42] = else spib_dma_req[4] GPCFG6[14] = 1 : dma_req[43] = rti3 dma req[1] ,0 : GPCFG6[14] = 0 : dma_req[43] = else spib_dma_req[5] GPCFG6[26] = 1 : irq_req[108] = rti2_int_req[0] ,0 : GPCFG6[26] = 0 : irq_req[108] = else epwm2_int1 GPCFG6[27] = 1 : irq_req[109] = rti2_int_req[1] ,0 : GPCFG6[27] = 0 : irq_req[109] = else epwm2_int2 GPCFG6[28] = 1 : irq_req[110] = rti3_int_req[0] ,0 : GPCFG6[28] = 0 : irq_req[110] = else epwm3_int1 GPCFG6[29] = 1 : irq_req[111] = rti3_int_req[1] ,0 : GPCFG6[29] = 0 : irq_req[111] = else epwm3_int2

4.7.3.7 GPCFG11 Register (Offset = 2Ch) [reset = 0h]

GPCFG11 is shown in Figure 4-858 and described in Table 4-899.

Return to Summary Table.

Figure 4-274. GPCFG11 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED						BSS2DSSSWIRQ2	BSS2DSSSWIRQ1
R-0h						0h	0h
15	14	13	12	11	10	9	8
RESERVED						DSS2BSSSWIRQ2	DSS2BSSSWIRQ1
R-0h						0h	0h
7	6	5	4	3	2	1	0
RESERVED						MSS2BSSSWIRQ2	MSS2BSSSWIRQ1
R-0h						0h	0h

Table 4-296. GPCFG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED		0h	Reserved
17	BSS2DSSSWIRQ2		0h	Self clearing register bit to generate interrupt to DSP from BSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
16	BSS2DSSSWIRQ1		0h	Self clearing register bit to generate interrupt to DSP from BSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
15-10	RESERVED	R	0h	Reserved
9	DSS2BSSSWIRQ2		0h	Self clearing register bit to generate interrupt to BSS from DSP. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
8	DSS2BSSSWIRQ1		0h	Self clearing register bit to generate interrupt to BSS from DSP. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
7-2	RESERVED	R	0h	Reserved
1	MSS2BSSSWIRQ2		0h	Self clearing register bit to generate interrupt to BSS from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
0	MSS2BSSSWIRQ1		0h	Self clearing register bit to generate interrupt to BSS from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS

4.7.3.8 CCCACFG0 Register (Offset = D0h) [reset = 0h]

CCCACFG0 is shown in Figure 4-859 and described in Table 4-900.

Return to Summary Table.

Figure 4-275. CCCACFG0 Register

31	30	29	28	27	26	25	24
CCCA_MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
CCCA_MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU31							CCCA_SINGLE_SHOT_MODE
0h							R/W-0h
7	6	5	4	3	2	1	0
CCCA_ENABLE_MODULE	CCCA_DISABLE_CLOCKS	CCCA_CLOCK1_SEL			CCCA_CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-297. CCCACFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CCCA_MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU31		0h	Reserved
8	CCCA_SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	CCCA_ENABLE_MODULE	R/W	0h	Enable for CCC
6	CCCA_DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CCCA_CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CCCA_CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.7.3.9 CCCACFG1 Register (Offset = D4h) [reset = 0h]

CCCACFG1 is shown in Figure 4-860 and described in Table 4-901.

Return to Summary Table.

Figure 4-276. CCCACFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG1																															
R/W-0h																															

Table 4-298. CCCACFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG1	R/W	0h	count0_expiry_val

4.7.3.10 CCCACFG2 Register (Offset = D8h) [reset = 0h]

CCCACFG2 is shown in Figure 4-861 and described in Table 4-902.

Return to Summary Table.

Figure 4-277. CCCACFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG2																															
R/W-0h																															

Table 4-299. CCCACFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG2	R/W	0h	count1_expected_val

4.7.3.11 CCCACFG3 Register (Offset = DCh) [reset = 0h]

CCCACFG3 is shown in Figure 4-862 and described in Table 4-903.

Return to Summary Table.

Figure 4-278. CCCACFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG3																															
R/W-0h																															

Table 4-300. CCCACFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG3	R/W	0h	Error Counter value in counter1 clock

4.7.3.12 CCCBCFG0 Register (Offset = E0h) [reset = 0h]

CCCBCFG0 is shown in Figure 4-863 and described in Table 4-904.

Return to Summary Table.

Figure 4-279. CCCBCFG0 Register

31	30	29	28	27	26	25	24
CCCB_MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
CCCB_MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU32							CCCB_SINGLE_SHOT_MODE
0h							R/W-0h
7	6	5	4	3	2	1	0
CCCB_ENABLE_MODULE	CCCB_DISABLE_CLOCKS	CCCB_CLOCK1_SEL			CCCB_CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-301. CCCBCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CCCB_MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU32		0h	Reserved
8	CCCB_SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	CCCB_ENABLE_MODULE	R/W	0h	Enable for CCC
6	CCCB_DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CCCB_CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CCCB_CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.7.3.13 CCCBCFG1 Register (Offset = E4h) [reset = 0h]

CCCBCFG1 is shown in Figure 4-864 and described in Table 4-905.

Return to Summary Table.

Figure 4-280. CCCBCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG1																															
R/W-0h																															

Table 4-302. CCCBCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG1	R/W	0h	count0_expiry_val

4.7.3.14 CCCBCFG2 Register (Offset = E8h) [reset = 0h]

CCCBCFG2 is shown in Figure 4-865 and described in Table 4-906.

Return to Summary Table.

Figure 4-281. CCCBCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG2																															
R/W-0h																															

Table 4-303. CCCBCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG2	R/W	0h	count1_expected_val

4.7.3.15 CCCBCFG3 Register (Offset = ECh) [reset = 0h]

CCCBCFG3 is shown in Figure 4-866 and described in Table 4-907.

Return to Summary Table.

Figure 4-282. CCCBCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG3																															
R/W-0h																															

Table 4-304. CCCBCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG3	R/W	0h	Error Counter value in counter1 clock

4.7.3.16 CCCACNTVAL Register (Offset = F0h) [reset = 0h]

CCCACNTVAL is shown in Figure 4-867 and described in Table 4-908.

Return to Summary Table.

Figure 4-283. CCCACNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACNTVAL																															
R-0h																															

Table 4-305. CCCACNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACNTVAL	R	0h	count1_val_out

4.7.3.17 CCCBCNTVAL Register (Offset = F4h) [reset = 0h]

CCCBCNTVAL is shown in Figure 4-868 and described in Table 4-909.

Return to Summary Table.

Figure 4-284. CCCBCNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCNTVAL																															
R-0h																															

Table 4-306. CCCBCNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCNTVAL	R	0h	count1_val_out

4.7.3.18 CCCABERRSTAT Register (Offset = F8h) [reset = 0h]

CCCABERRSTAT is shown in Figure 4-869 and described in Table 4-910.

Return to Summary Table.

Figure 4-285. CCCABERRSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCABERRSTAT																															
R-0h																															

Table 4-307. CCCABERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCABERRSTAT	R	0h	7:0 : CCCA Error Status 15:8 : CCCB Error Status

4.7.3.19 USERMODEEN Register (Offset = FCh) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-286. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-308. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS GPCFG space.

4.7.3.20 EPWMCFG Register (Offset = 140h) [reset = F00h]

EPWMCFG is shown in Figure 4-871 and described in Table 4-912.

Return to Summary Table.

Figure 4-287. EPWMCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPWMCFG																															
R/W-F00h																															

Table 4-309. EPWMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EPWMCFG	R/W	F00h	[1:0] : 0->rampgen Frame start is SYNCIN to EPWM1 1->FRC Frame Start is SYNCIN to EPWM1 2,3-> External SYNCIN is SYNCIN to EPWM1 [3:2]: 0->rampgen Frame start is SYNCIN to EPWM2 1->FRC Frame Start is SYNCIN to EPWM2 2,3-> EPWM1 SYNCO is SYNCIN to EPWM2 [5:4]: 0->rampgen Frame start is SYNCIN to EPWM3 1->FRC Frame Start is SYNCIN to EPWM3 2,3-> EPWM2 SYNCO is SYNCIN to EPWM3 [31:8] : Reserved

4.7.3.21 DMMSWINT0 Register (Offset = 148h) [reset = 0h]

DMMSWINT0 is shown in Figure 4-872 and described in Table 4-913.

Return to Summary Table.

Figure 4-288. DMMSWINT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINT0																															
0h																															

Table 4-310. DMMSWINT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINT0		0h	HIL Intr0 muxed with frame start interrupt to DSP HIL Intr1 muxed with logical frame start interrupt to DSP HIL Intr2 muxed with Ping/Pong threshold trigger interrupt for DSP HIL Intr3 muxed with ADC data valid interrupt for DSP HIL Intr4 - HIL Intr8 : SW interrupt for DSP HIL Intr9 muxed with frame start interrupt to TPCC0 HIL Intr10 muxed with logical frame start interrupt to TPCC0 HIL Intr11 muxed with Ping/Pong threshold trigger interrupt for TPCC0 HIL Intr12 muxed with ADC data valid interrupt for TPCC0 HIL Intr13 - HIL Intr18 as SW interrupt for TPCC0 HIL Intr19 muxed with frame start interrupt to TPCC1 HIL Intr20 muxed with logical frame start interrupt to TPCC1 HIL Intr21 muxed with Ping/Pong threshold trigger interrupt for TPCC1 HIL Intr22 muxed with ADC data valid interrupt for TPCC1 HIL Intr23 - HIL Intr28 as SW interrupt for TPCC1 HIL Intr29 muxed with frame start interrupt to VIMMR4F HIL Intr30 muxed with logical frame start interrupt to VIMMR4F HIL Intr31 muxed with Ping/Pong threshold trigger interrupt for VIMMR4F HIL Intr32 muxed with ADC data valid interrupt for VIMMR4F HIL Intr33 - HIL Intr38 as SW interrupt for VIMMR4F HIL Intr39 muxed with frame start interrupt right at the source - propagates throughout the system as actual FRC output does. This is muxed with the frame start coming from BSS to TOP so that it propagates to MSS/DSS. HIL Intr40 muxed with frame start interrupt right at the source - propagates throughout the system as actual FRC output does. This is muxed with the ADC clock enable signal coming from FRC. Intr41 muxed with logical frame start interrupt right at the source (FRC) - propagates throughout the system Intr42 muxed with logical frame end interrupt right at the source (FRC) - propagates throughout the system Intr43 muxed with Ping/Pong threshold trigger interrupt right at the source - propagates throughout the system Intr44 muxed with ADC data valid interrupt right at the source - propagates throughout the system

4.7.3.22 DMMSWINT1 Register (Offset = 14Ch) [reset = 0h]

DMMSWINT1 is shown in Figure 4-953 and described in Table 4-996.

Return to Summary Table.

Figure 4-289. DMMSWINT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINT1																															
0h																															

Table 4-311. DMMSWINT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINT1		0h	HIL Interrupts - MSB 32 bits [63:32]

4.7.3.23 DMMSWINTSEL0 Register (Offset = 150h) [reset = 0h]

DMMSWINTSEL0 is shown in Figure 4-874 and described in Table 4-915.

Return to Summary Table.

Figure 4-290. DMMSWINTSEL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINTSEL0																															
R/W-0h																															

Table 4-312. DMMSWINTSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINTSEL0	R/W	0h	Mux control for HIL Interrupts - LSB 32 bits

4.7.3.24 DMMSWINTSEL1 Register (Offset = 154h) [reset = 0h]

DMMSWINTSEL1 is shown in Figure 4-875 and described in Table 4-916.

Return to Summary Table.

Figure 4-291. DMMSWINTSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINTSEL1																															
R/W-0h																															

Table 4-313. DMMSWINTSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINTSEL1	R/W	0h	Mux control for HIL Interrupts - MSB 32 bits

4.7.3.25 CCCBWDEN Register (Offset = 158h) [reset = 0h]

CCCBWDEN is shown in Figure 4-876 and described in Table 4-917.

Return to Summary Table.

Figure 4-292. CCCBWDEN Register

31	30	29	28	27	26	25	24
NU36							
R-0h							
23	22	21	20	19	18	17	16
NU36							ENABLECCBERRRSTN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU35							
R-0h							
7	6	5	4	3	2	1	0
NU35							ENABLECCBERRNMI
R-0h							R/W-0h

Table 4-314. CCCBWDEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU36	R	0h	Reserved
16	ENABLECCBERRRSTN	R/W	0h	Enable CCCB error to generate WD restn.In this mode CCCB has to compare MSS CR4 clock to XTAL
15-1	NU35	R	0h	Reserved
0	ENABLECCBERRNMI	R/W	0h	Enable CCCB error to generate NMI.In this mode CCCB has to compare MSS CR4 clock to XTAL

4.7.3.26 GPIOINTREDGESEL Register (Offset = 15Ch) [reset = 0h]

GPIOINTREDGESEL is shown in Figure 4-877 and described in Table 4-918.

Return to Summary Table.

Figure 4-293. GPIOINTREDGESEL Register

31	30	29	28	27	26	25	24
NU37							
0h							
23	22	21	20	19	18	17	16
NU37							
0h							
15	14	13	12	11	10	9	8
NU37							
0h							
7	6	5	4	3	2	1	0
NU37					GPIO2EDGESEL	GPIO1EDGESEL	GPIO0EDGESEL
0h					R/W-0h	R/W-0h	R/W-0h

Table 4-315. GPIOINTREDGESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU37		0h	Reserved
2	GPIO2EDGESEL	R/W	0h	0:Posedge of GPIO2 generates interruptt to MSS CR4 and DSP 1:Negedge
1	GPIO1EDGESEL	R/W	0h	0:Posedge of GPIO1 generates interruptt to MSS CR4 and DSP 1:Negedge
0	GPIO0EDGESEL	R/W	0h	0:Posedge of GPIO0 generates interruptt to MSS CR4 and DSP 1:Negedge

4.7.3.27 PWMDMATRIGEN Register (Offset = 164h) [reset = 0h]

PWMDMATRIGEN is shown in Figure 4-878 and described in Table 4-919.

Return to Summary Table.

Figure 4-294. PWMDMATRIGEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU38															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU38												PWMDMATRIGEN			
R-0h												R/W-0h			

Table 4-316. PWMDMATRIGEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU38	R	0h	Reserved
3-0	PWMDMATRIGEN	R/W	0h	0: "1" -> Mux epwm1a instead of mss_event_gen_1_frc to DMA 1: "1" -> Mux epwm1b instead of mss_event_gen_1_frc to DMA 2: "1" -> Mux epwm2a instead of mss_event_gen_1_frc to DMA 3: "1" -> Mux epwm2b instead of mss_event_gen_1_frc to DMA

4.7.3.28 JTAGTXDATA Register (Offset = 168h) [reset = 0h]

JTAGTXDATA is shown in Figure 4-879 and described in Table 4-920.

Return to Summary Table.

Figure 4-295. JTAGTXDATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGTXDATA																															
R-0h																															

Table 4-317. JTAGTXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGTXDATA	R	0h	Transmit Data. This register is used to pass data to the system security logic

4.7.3.29 JTAGTXCONTROL Register (Offset = 16Ch) [reset = 0h]

JTAGTXCONTROL is shown in Figure 4-880 and described in Table 4-921.

Return to Summary Table.

Figure 4-296. JTAGTXCONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGTXCONTROL																															
R-0h																															

Table 4-318. JTAGTXCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGTXCONTROL	R	0h	Provides the handshake for the JTAGTXDATA Register and can also be used to pass control information to the system security logic. Only bits [31:1] are valid

4.7.3.30 JTAGRXDATA Register (Offset = 170h) [reset = 0h]

JTAGRXDATA is shown in Figure 4-881 and described in Table 4-922.

Return to Summary Table.

Figure 4-297. JTAGRXDATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGRXDATA																															
R/W-0h																															

Table 4-319. JTAGRXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGRXDATA	R/W	0h	Receive data. This register is used to pass data from the system security logic

4.7.3.31 JTAGTXRXACK Register (Offset = 178h) [reset = 0h]

JTAGTXRXACK is shown in Figure 4-882 and described in Table 4-923.

Return to Summary Table.

Figure 4-298. JTAGTXRXACK Register

31	30	29	28	27	26	25	24
NU40							
R-0h							
23	22	21	20	19	18	17	16
NU40							
R-0h							
15	14	13	12	11	10	9	8
NU40							JTAGRXDATAWR
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NU39							JTAGTXDATARD
R-0h							R/W-0h

Table 4-320. JTAGTXRXACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	NU40	R	0h	Reserved
8	JTAGRXDATAWR	R/W	0h	Indication from system security logic that JTAGRXDATA is valid
7-1	NU39	R	0h	Reserved
0	JTAGTXDATARD	R/W	0h	Indication from the system security logic that JTAGTXDATA has been accepted

4.7.3.32 JTAGRXCONTROL Register (Offset = 17Ch) [reset = 0h]

JTAGRXCONTROL is shown in Figure 4-883 and described in Table 4-924.

Return to Summary Table.

Figure 4-299. JTAGRXCONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGRXCONTROL																															
R/W-0h																															

Table 4-321. JTAGRXCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGRXCONTROL	R/W	0h	Provides the handshake for the JTAGRXDATA Register and can also be used to pass control information from the system security logic. Only bits [31:1] is used.

4.7.3.33 MSS2GEMSWIRQ Register (Offset = 180h) [reset = 0h]

MSS2GEMSWIRQ is shown in Figure 4-884 and described in Table 4-925.

Return to Summary Table.

Figure 4-300. MSS2GEMSWIRQ Register

31	30	29	28	27	26	25	24
NU41							
R-0h							
23	22	21	20	19	18	17	16
NU41							
R-0h							
15	14	13	12	11	10	9	8
NU41							
R-0h							
7	6	5	4	3	2	1	0
NU41						MSS2GEMSWIRQ2	MSS2GEMSWIRQ1
R-0h						0h	0h

Table 4-322. MSS2GEMSWIRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU41	R	0h	Reserved
1	MSS2GEMSWIRQ2		0h	Self clearing register bit to generate interrupt to DSP from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
0	MSS2GEMSWIRQ1		0h	Self clearing register bit to generate interrupt to DSP from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP

4.7.3.34 CSETBFLUSH Register (Offset = 184h) [reset = 0h]

CSETBFLUSH is shown in Figure 4-885 and described in Table 4-926.

Return to Summary Table.

Figure 4-301. CSETBFLUSH Register

31	30	29	28	27	26	25	24
NU43							
R-0h							
23	22	21	20	19	18	17	16
NU43							
R-0h							
15	14	13	12	11	10	9	8
NU43					CSETBFULL	CSETBACQ_OMPLETE	CSETBFLUSHINACK
R-0h					R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
NU42							CSETBFLUSHIN
R-0h							R/W-0h

Table 4-323. CSETBFLUSH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU43	R	0h	Reserved
10	CSETBFULL	R	0h	When HIGH indicates that the ETB RAM has overflowed or wrapped around to address zero
9	CSETBACQ_OMPLETE	R	0h	When HIGH, indicates that trace acquisition is complete by ETB, that is, the trigger counter is at zero
8	CSETBFLUSHINACK	R	0h	Return acknowledgement to CSETBFLUSHIN
7-1	NU42	R	0h	Reserved
0	CSETBFLUSHIN	R/W	0h	External control used to assert the ATB signal AFVALIDS and drain any historical FIFO information on the bus

4.7.4 DSS_REG Registers

Table 4-927 lists the memory-mapped registers for the DSS_REG. All register offset addresses not listed in Table 4-927 should be considered as reserved locations and the register contents should not be modified.

Table 4-324. DSS_REG Registers

Offset	Acronym	Register Name	Section
50h	RTIEVENTCAPTURESEL	RTIEVENTCAPTURESEL	Section 4.9.4.1
6Ch	CQCFG1	CQCFG1	Section 4.9.4.2
80h	TPCCPARSTATCFG	TPCCPARSTATCFG	Section 4.9.4.3
104h	TPTC0WRMPUSTADD0	TPTC0WRMPUSTADD0	Section 4.9.4.4
108h	TPTC0WRMPUSTADD1	TPTC0WRMPUSTADD1	Section 4.9.4.5
10Ch	TPTC0WRMPUSTADD2	TPTC0WRMPUSTADD2	Section 4.9.4.6
110h	TPTC0WRMPUSTADD3	TPTC0WRMPUSTADD3	Section 4.9.4.7
114h	TPTC0WRMPUSTADD4	TPTC0WRMPUSTADD4	Section 4.9.4.8
118h	TPTC0WRMPUSTADD5	TPTC0WRMPUSTADD5	Section 4.9.4.9
124h	TPTC0WRMPUENDADD0	TPTC0WRMPUENDADD0	Section 4.9.4.10
128h	TPTC0WRMPUENDADD1	TPTC0WRMPUENDADD1	Section 4.9.4.11
12Ch	TPTC0WRMPUENDADD2	TPTC0WRMPUENDADD2	Section 4.9.4.12
130h	TPTC0WRMPUENDADD3	TPTC0WRMPUENDADD3	Section 4.9.4.13
134h	TPTC0WRMPUENDADD4	TPTC0WRMPUENDADD4	Section 4.9.4.14
138h	TPTC0WRMPUENDADD5	TPTC0WRMPUENDADD5	Section 4.9.4.15
144h	TPTC0WRMPUERRADD	TPTC0WRMPUERRADD	Section 4.9.4.16
148h	TPTC0RDMPUSTADD0	TPTC0RDMPUSTADD0	Section 4.9.4.17
14Ch	TPTC0RDMPUSTADD1	TPTC0RDMPUSTADD1	Section 4.9.4.18
150h	TPTC0RDMPUSTADD2	TPTC0RDMPUSTADD2	Section 4.9.4.19
154h	TPTC0RDMPUSTADD3	TPTC0RDMPUSTADD3	Section 4.9.4.20
158h	TPTC0RDMPUSTADD4	TPTC0RDMPUSTADD4	Section 4.9.4.21
15Ch	TPTC0RDMPUSTADD5	TPTC0RDMPUSTADD5	Section 4.9.4.22
168h	TPTC0RDMPUENDADD0	TPTC0RDMPUENDADD0	Section 4.9.4.23
16Ch	TPTC0RDMPUENDADD1	TPTC0RDMPUENDADD1	Section 4.9.4.24
170h	TPTC0RDMPUENDADD2	TPTC0RDMPUENDADD2	Section 4.9.4.25
174h	TPTC0RDMPUENDADD3	TPTC0RDMPUENDADD3	Section 4.9.4.26
178h	TPTC0RDMPUENDADD4	TPTC0RDMPUENDADD4	Section 4.9.4.27
17Ch	TPTC0RDMPUENDADD5	TPTC0RDMPUENDADD5	Section 4.9.4.28
188h	TPTC0RDMPUERRADD	TPTC0RDMPUERRADD	Section 4.9.4.29
18Ch	TPTC1WRMPUSTADD0	TPTC1WRMPUSTADD0	Section 4.9.4.30
190h	TPTC1WRMPUSTADD1	TPTC1WRMPUSTADD1	Section 4.9.4.31
194h	TPTC1WRMPUSTADD2	TPTC1WRMPUSTADD2	Section 4.9.4.32
198h	TPTC1WRMPUSTADD3	TPTC1WRMPUSTADD3	Section 4.9.4.33
19Ch	TPTC1WRMPUSTADD4	TPTC1WRMPUSTADD4	Section 4.9.4.34
1A0h	TPTC1WRMPUSTADD5	TPTC1WRMPUSTADD5	Section 4.9.4.35
1ACh	TPTC1WRMPUENDADD0	TPTC1WRMPUENDADD0	Section 4.9.4.36
1B0h	TPTC1WRMPUENDADD1	TPTC1WRMPUENDADD1	Section 4.9.4.37
1B4h	TPTC1WRMPUENDADD2	TPTC1WRMPUENDADD2	Section 4.9.4.38
1B8h	TPTC1WRMPUENDADD3	TPTC1WRMPUENDADD3	Section 4.9.4.39
1BCh	TPTC1WRMPUENDADD4	TPTC1WRMPUENDADD4	Section 4.9.4.40
1C0h	TPTC1WRMPUENDADD5	TPTC1WRMPUENDADD5	Section 4.9.4.41
1CCh	TPTC1WRMPUERRADD	TPTC1WRMPUERRADD	Section 4.9.4.42
1D0h	TPTC1RDMPUSTADD0	TPTC1RDMPUSTADD0	Section 4.9.4.43

Table 4-324. DSS_REG Registers (continued)

Offset	Acronym	Register Name	Section
1D4h	TPTC1RDMPUSTADD1	TPTC1RDMPUSTADD1	Section 4.9.4.44
1D8h	TPTC1RDMPUSTADD2	TPTC1RDMPUSTADD2	Section 4.9.4.45
1DCh	TPTC1RDMPUSTADD3	TPTC1RDMPUSTADD3	Section 4.9.4.46
1E0h	TPTC1RDMPUSTADD4	TPTC1RDMPUSTADD4	Section 4.9.4.47
1E4h	TPTC1RDMPUSTADD5	TPTC1RDMPUSTADD5	Section 4.9.4.48
1F0h	TPTC1RDMPUENDADD0	TPTC1RDMPUENDADD0	Section 4.9.4.49
1F4h	TPTC1RDMPUENDADD1	TPTC1RDMPUENDADD1	Section 4.9.4.50
1F8h	TPTC1RDMPUENDADD2	TPTC1RDMPUENDADD2	Section 4.9.4.51
1FCh	TPTC1RDMPUENDADD3	TPTC1RDMPUENDADD3	Section 4.9.4.52
200h	TPTC1RDMPUENDADD4	TPTC1RDMPUENDADD4	Section 4.9.4.53
204h	TPTC1RDMPUENDADD5	TPTC1RDMPUENDADD5	Section 4.9.4.54
210h	TPTC1RDMPUERRADD	TPTC1RDMPUERRADD	Section 4.9.4.55
214h	TPTCMPUVALIDCFG	TPTCMPUVALIDCFG	Section 4.9.4.56
218h	TPTCMPUENCFG	TPTCMPUENCFG	Section 4.9.4.57
21Ch	TESTPATTERNRX1ICFG	TESTPATTERNRX1ICFG	Section 4.9.4.58
220h	TESTPATTERNRX2ICFG	TESTPATTERNRX2ICFG	Section 4.9.4.59
224h	TESTPATTERNRX3ICFG	TESTPATTERNRX3ICFG	Section 4.9.4.60
228h	TESTPATTERNRX4ICFG	TESTPATTERNRX4ICFG	Section 4.9.4.61
22Ch	TESTPATTERNRX1QCFG	TESTPATTERNRX1QCFG	Section 4.9.4.62
230h	TESTPATTERNRX2QCFG	TESTPATTERNRX2QCFG	Section 4.9.4.63
234h	TESTPATTERNRX3QCFG	TESTPATTERNRX3QCFG	Section 4.9.4.64
238h	TESTPATTERNRX4QCFG	TESTPATTERNRX4QCFG	Section 4.9.4.65
23Ch	TESTPATTERNVLDCFG	TESTPATTERNVLDCFG	Section 4.9.4.66
240h	DSSMISC	DSSMISC	
258h	TPCC1PARSTATCFG	TPCC1PARSTATCFG	Section 4.9.4.67
260h	DMMSWINT1	DMMSWINT1	Section 4.9.4.68
270h	DSSINTRCFG	DSSINTRCFG	Section 4.9.4.69
274h	MPUMSTIDCFG1	MPUMSTIDCFG1	Section 4.9.4.70
278h	MPUMSTIDCFG2	MPUMSTIDCFG2	Section 4.9.4.71
27Ch	MPUMSTIDCFG3	MPUMSTIDCFG3	Section 4.9.4.72
280h	HSRAM1ECCCFG	HSRAM1ECCCFG	Section 4.9.4.73
288h	DATATRRAMECCCFG	DATATRRAMECCCFG	Section 4.9.4.74
28Ch	ADCBUFPINGECCCFG	ADCBUFPINGECCCFG	Section 4.9.4.75
290h	ADCBUFPONGECCCFG	ADCBUFPONGECCCFG	Section 4.9.4.76
29Ch	UMAP0PARITYCFG1	UMAP0PARITYCFG1	Section 4.9.4.77
2A0h	UMAP0PARITYCFG2	UMAP0PARITYCFG2	Section 4.9.4.78
2A4h	UMAP0PARITYCFG3	UMAP0PARITYCFG3	Section 4.9.4.79
2A8h	UMAP1PARITYCFG1	UMAP1PARITYCFG1	Section 4.9.4.80
2ACh	UMAP1PARITYCFG2	UMAP1PARITYCFG2	Section 4.9.4.81
2B0h	UMAP1PARITYCFG3	UMAP1PARITYCFG3	Section 4.9.4.82
2B4h	ESMGRP2MASKCFG	ESMGRP2MASKCFG	Section 4.9.4.83
2B8h	L2MEMINITCFG1[(1)]	L2MEMINITCFG1	Section 4.9.4.84
2BCh	L2MEMINITCFG2[(1)]	L2MEMINITCFG2	Section 4.9.4.85
2C0h	GEMRSTCAUSE[(1)]	GEMRSTCAUSE	Section 4.9.4.86
2CCh	GEMPWRSMCFG4[(1)]	GEMPWRSMCFG4	Section 4.9.4.87
2D4h	PWRSMWAKEMASK0[(1)]	PWRSMWAKEMASK0	Section 4.9.4.88

(1) Applicable for devices with DSP C674x (such as the 68xx device).

Table 4-324. DSS_REG Registers (continued)

Offset	Acronym	Register Name	Section
2D8h	PWRSMWAKEMASK1[(1)]	PWRSMWAKEMASK1	Section 4.9.4.89
2DCh	PWRSMWAKEMASK2[(1)]	PWRSMWAKEMASK2	Section 4.9.4.90
2E0h	PWRSMMISEVTMASK0[(1)]	PWRSMMISEVTMASK0	Section 4.9.4.91
2E4h	PWRSMMISEVTMASK1[(1)]	PWRSMMISEVTMASK1	Section 4.9.4.92
2E8h	PWRSMMISEVTMASK2[(1)]	PWRSMMISEVTMASK2	Section 4.9.4.93
2ECh	PWRSMWAKESRCSTAT0[(1)]	PWRSMWAKESRCSTAT0	Section 4.9.4.94
2F0h	PWRSMWAKESRCSTAT1[(1)]	PWRSMWAKESRCSTAT1	Section 4.9.4.95
320h	PWRSMWAKESRCSTAT2[(1)]	PWRSMWAKESRCSTAT2	Section 4.9.4.96
324h	PWRSMEVNTMONSTAT0[(1)]	PWRSMEVNTMONSTAT0	Section 4.9.4.97
328h	PWRSMEVNTMONSTAT1[(1)]	PWRSMEVNTMONSTAT1	Section 4.9.4.98
32Ch	PWRSMEVNTMONSTAT2[(1)]	PWRSMEVNTMONSTAT2	Section 4.9.4.99
330h	PWRSMWAKESRCSTATCLR0[(1)]	PWRSMWAKESRCSTATCLR0	Section 4.9.4.100
334h	PWRSMWAKESRCSTATCLR1[(1)]	PWRSMWAKESRCSTATCLR1	Section 4.9.4.101
338h	PWRSMWAKESRCSTATCLR2[(1)]	PWRSMWAKESRCSTATCLR2	Section 4.9.4.102
33Ch	ADCBUFCFG1	ADCBUFCFG1	Section 4.9.4.103
340h	ADCBUFCFG2	ADCBUFCFG2	Section 4.9.4.104
344h	ADCBUFCFG3	ADCBUFCFG3	Section 4.9.4.105
348h	ADCBUFCFG4	ADCBUFCFG4	Section 4.9.4.106
34Ch	STCPBISTSMCFG1	STCPBISTSMCFG1	Section 4.9.4.107
350h	STCPBISTSMCFG2	STCPBISTSMCFG2	Section 4.9.4.108
358h	RTI2EVENTCAPTURESEL	RTI2EVENTCAPTURESEL	Section 4.9.4.109
35Ch	DSSMISC5	DSSMISC5	Section 4.9.4.110

Complex bit access types are encoded to fit into small table cells. Table 4-928 shows the codes that are used for access types in this section.

Table 4-325. DSS_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.7.4.1 RTIEVENTCAPTURESEL Register (Offset = 50h) [reset = 0h]

RTIEVENTCAPTURESEL is shown in Figure 4-886 and described in Table 4-929.

Return to Summary Table.

Figure 4-302. RTIEVENTCAPTURESEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2									EVT1							NU1									EVT0							
R-0h									R/W-0h							R-0h									R/W-0h						

Table 4-326. RTIEVENTCAPTURESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	
22-16	EVT1	R/W	0h	Setting the source of interrupt for Counter value capture for RTI1 Event1
15-7	NU1	R	0h	
6-0	EVT0	R/W	0h	Setting the source of interrupt for Counter value capture for RTI1 Event0.

4.7.4.2 CQCFG1 Register (Offset = 6Ch) [reset = 40100000h]

CQCFG1 is shown in Figure 4-887 and described in Table 4-930.

Return to Summary Table.

Figure 4-303. CQCFG1 Register

31	30	29	28	27	26	25	24
NU3	CQ2BASEADDR						
R-0h	R/W-100h						
23	22	21	20	19	18	17	16
CQ2BASEADDR			CQ1BASEADDR				
R/W-100h			R/W-80h				
15	14	13	12	11	10	9	8
CQ1BASEADDR			CQ0BASEADDR				
R/W-80h			R/W-0h				
7	6	5	4	3	2	1	0
CQ0BASEADDR				CQ96BITPACKEN	NU	CQDATAWIDTH	
R/W-0h				R/W-0h	R-0h	R/W-0h	

Table 4-327. CQCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU3	R	0h	
30-22	CQ2BASEADDR	R/W	100h	128-bit Address offset which inidcates the start address for storing CQ0 (ADC/RxIF Saturation Detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
21-13	CQ1BASEADDR	R/W	80h	128-bit Address offset which inidcates the start address for storing CQ0 (Signal Image Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
12-4	CQ0BASEADDR	R/W	0h	128-bit Address offset which inidcates the start address for storing CQ0 (Wide Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
3	CQ96BITPACKEN	R/W	0h	This is used to pack the CQ data into only the LSB 96 bits of each row of the CQ memory. This can be used in 3 channel mode of LVDS where the ADC data and Chirp Params occupy only LSB 96 bits of each memory row.
2	NU	R	0h	
1-0	CQDATAWIDTH	R/W	0h	This is used to appropriately pack the valid CQ data bits in appropriate bits in the CQ memory. 00, 01->Raw 16, 10-->Raw 12, 11-->Raw14

4.7.4.3 TPCCPARSTATCFG Register (Offset = 80h) [reset = 0h]

TPCCPARSTATCFG is shown in Figure 4-888 and described in Table 4-931.

Return to Summary Table.

Figure 4-304. TPCCPARSTATCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU					TPCCPARITYTSTEN	TPCCPARITYEN	TPCCPARITYCLR
R-0h					R/W-0h	R/W-0h	0h
7	6	5	4	3	2	1	0
TPCCPARITYSTAT							
R-0h							

Table 4-328. TPCCPARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	
10	TPCCPARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in TPCC
9	TPCCPARITYEN	R/W	0h	Enable bit for the Parity computation in TPCC
8	TPCCPARITYCLR		0h	Write 0x1 to clear the status. This is a special access type; a write to this field generates a pulse.
7-0	TPCCPARITYSTAT	R	0h	Parity address from TPCC

4.7.4.4 TPTC0WRMPUSTADD0 Register (Offset = 104h) [reset = 0h]

TPTC0WRMPUSTADD0 is shown in Figure 4-889 and described in Table 4-932.

Return to Summary Table.

Figure 4-305. TPTC0WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD0																															
R/W-0h																															

Table 4-329. TPTC0WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC0

4.7.4.5 TPTC0WRMPUSTADD1 Register (Offset = 108h) [reset = 0h]

TPTC0WRMPUSTADD1 is shown in Figure 4-890 and described in Table 4-933.

Return to Summary Table.

Figure 4-306. TPTC0WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD1																															
R/W-0h																															

Table 4-330. TPTC0WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC0

4.7.4.6 TPTC0WRMPUSTADD2 Register (Offset = 10Ch) [reset = 0h]

TPTC0WRMPUSTADD2 is shown in Figure 4-891 and described in Table 4-934.

Return to Summary Table.

Figure 4-307. TPTC0WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD2																															
R/W-0h																															

Table 4-331. TPTC0WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC0

4.7.4.7 TPTC0WRMPUSTADD3 Register (Offset = 110h) [reset = 0h]

TPTC0WRMPUSTADD3 is shown in Figure 4-892 and described in Table 4-935.

Return to Summary Table.

Figure 4-308. TPTC0WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD3																															
R/W-0h																															

Table 4-332. TPTC0WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC0

4.7.4.8 TPTC0WRMPUSTADD4 Register (Offset = 114h) [reset = 0h]

TPTC0WRMPUSTADD4 is shown in Figure 4-893 and described in Table 4-936.

Return to Summary Table.

Figure 4-309. TPTC0WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD4																															
R/W-0h																															

Table 4-333. TPTC0WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC0

4.7.4.9 TPTC0WRMPUSTADD5 Register (Offset = 118h) [reset = 0h]

TPTC0WRMPUSTADD5 is shown in Figure 4-894 and described in Table 4-937.

Return to Summary Table.

Figure 4-310. TPTC0WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD5																															
R/W-0h																															

Table 4-334. TPTC0WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC0

4.7.4.10 TPTC0WRMPUENDADD0 Register (Offset = 124h) [reset = 0h]

TPTC0WRMPUENDADD0 is shown in Figure 4-895 and described in Table 4-938.

Return to Summary Table.

Figure 4-311. TPTC0WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD0																															
R/W-0h																															

Table 4-335. TPTC0WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC0

4.7.4.11 TPTC0WRMPUENDADD1 Register (Offset = 128h) [reset = 0h]

TPTC0WRMPUENDADD1 is shown in Figure 4-896 and described in Table 4-939.

Return to Summary Table.

Figure 4-312. TPTC0WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD1																															
R/W-0h																															

Table 4-336. TPTC0WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC0

4.7.4.12 TPTC0WRMPUENDADD2 Register (Offset = 12Ch) [reset = 0h]

TPTC0WRMPUENDADD2 is shown in Figure 4-897 and described in Table 4-940.

Return to Summary Table.

Figure 4-313. TPTC0WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD2																															
R/W-0h																															

Table 4-337. TPTC0WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC0

4.7.4.13 TPTC0WRMPUENDADD3 Register (Offset = 130h) [reset = 0h]

TPTC0WRMPUENDADD3 is shown in Figure 4-898 and described in Table 4-941.

Return to Summary Table.

Figure 4-314. TPTC0WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD3																															
R/W-0h																															

Table 4-338. TPTC0WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC0

4.7.4.14 TPTC0WRMPUENDADD4 Register (Offset = 134h) [reset = 0h]

TPTC0WRMPUENDADD4 is shown in Figure 4-899 and described in Table 4-942.

Return to Summary Table.

Figure 4-315. TPTC0WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD4																															
R/W-0h																															

Table 4-339. TPTC0WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC0

4.7.4.15 TPTC0WRMPUENDADD5 Register (Offset = 138h) [reset = 0h]

TPTC0WRMPUENDADD5 is shown in Figure 4-900 and described in Table 4-943.

Return to Summary Table.

Figure 4-316. TPTC0WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD5																															
R/W-0h																															

Table 4-340. TPTC0WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC0

4.7.4.16 TPTC0WRMPUERRADD Register (Offset = 144h) [reset = 0h]

TPTC0WRMPUERRADD is shown in Figure 4-901 and described in Table 4-944.

Return to Summary Table.

Figure 4-317. TPTC0WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUERRADD																															
R-0h																															

Table 4-341. TPTC0WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC0

4.7.4.17 TPTC0RDMPUSTADD0 Register (Offset = 148h) [reset = 0h]

TPTC0RDMPUSTADD0 is shown in Figure 4-902 and described in Table 4-945.

Return to Summary Table.

Figure 4-318. TPTC0RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD0																															
R/W-0h																															

Table 4-342. TPTC0RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC0

4.7.4.18 TPTC0RDMPUSTADD1 Register (Offset = 14Ch) [reset = 0h]

TPTC0RDMPUSTADD1 is shown in Figure 4-903 and described in Table 4-946.

Return to Summary Table.

Figure 4-319. TPTC0RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD1																															
R/W-0h																															

Table 4-343. TPTC0RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC0

4.7.4.19 TPTC0RDMPUSTADD2 Register (Offset = 150h) [reset = 0h]

TPTC0RDMPUSTADD2 is shown in Figure 4-904 and described in Table 4-947.

Return to Summary Table.

Figure 4-320. TPTC0RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD2																															
R/W-0h																															

Table 4-344. TPTC0RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC0

4.7.4.20 TPTC0RDMPUSTADD3 Register (Offset = 154h) [reset = 0h]

TPTC0RDMPUSTADD3 is shown in Figure 4-905 and described in Table 4-948.

Return to Summary Table.

Figure 4-321. TPTC0RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD3																															
R/W-0h																															

Table 4-345. TPTC0RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC0

4.7.4.21 TPTC0RDMPUSTADD4 Register (Offset = 158h) [reset = 0h]

TPTC0RDMPUSTADD4 is shown in Figure 4-906 and described in Table 4-949.

Return to Summary Table.

Figure 4-322. TPTC0RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD4																															
R/W-0h																															

Table 4-346. TPTC0RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC0

4.7.4.22 TPTC0RDMPUSTADD5 Register (Offset = 15Ch) [reset = 0h]

TPTC0RDMPUSTADD5 is shown in Figure 4-907 and described in Table 4-950.

Return to Summary Table.

Figure 4-323. TPTC0RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD5																															
R/W-0h																															

Table 4-347. TPTC0RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC0

4.7.4.23 TPTC0RDMPUENDADD0 Register (Offset = 168h) [reset = 0h]

TPTC0RDMPUENDADD0 is shown in Figure 4-908 and described in Table 4-951.

Return to Summary Table.

Figure 4-324. TPTC0RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD0																															
R/W-0h																															

Table 4-348. TPTC0RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC0

4.7.4.24 TPTC0RDMPUENDADD1 Register (Offset = 16Ch) [reset = 0h]

TPTC0RDMPUENDADD1 is shown in Figure 4-909 and described in Table 4-952.

Return to Summary Table.

Figure 4-325. TPTC0RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD1																															
R/W-0h																															

Table 4-349. TPTC0RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC0

4.7.4.25 TPTC0RDMPUENDADD2 Register (Offset = 170h) [reset = 0h]

TPTC0RDMPUENDADD2 is shown in Figure 4-910 and described in Table 4-953.

Return to Summary Table.

Figure 4-326. TPTC0RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD2																															
R/W-0h																															

Table 4-350. TPTC0RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC0

4.7.4.26 TPTC0RDMPUENDADD3 Register (Offset = 174h) [reset = 0h]

TPTC0RDMPUENDADD3 is shown in Figure 4-911 and described in Table 4-954.

Return to Summary Table.

Figure 4-327. TPTC0RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD3																															
R/W-0h																															

Table 4-351. TPTC0RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC0

4.7.4.27 TPTC0RDMPUENDADD4 Register (Offset = 178h) [reset = 0h]

TPTC0RDMPUENDADD4 is shown in Figure 4-912 and described in Table 4-955.

Return to Summary Table.

Figure 4-328. TPTC0RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD4																															
R/W-0h																															

Table 4-352. TPTC0RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC0

4.7.4.28 TPTC0RDMPUENDADD5 Register (Offset = 17Ch) [reset = 0h]

TPTC0RDMPUENDADD5 is shown in Figure 4-913 and described in Table 4-956.

Return to Summary Table.

Figure 4-329. TPTC0RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD5																															
R/W-0h																															

Table 4-353. TPTC0RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC0

4.7.4.29 TPTC0RDMPUERRADD Register (Offset = 188h) [reset = 0h]

TPTC0RDMPUERRADD is shown in Figure 4-914 and described in Table 4-957.

Return to Summary Table.

Figure 4-330. TPTC0RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUERRADD																															
R-0h																															

Table 4-354. TPTC0RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC0

4.7.4.30 TPTC1WRMPUSTADD0 Register (Offset = 18Ch) [reset = 0h]

TPTC1WRMPUSTADD0 is shown in Figure 4-915 and described in Table 4-958.

Return to Summary Table.

Figure 4-331. TPTC1WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD0																															
R/W-0h																															

Table 4-355. TPTC1WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC1

4.7.4.31 TPTC1WRMPUSTADD1 Register (Offset = 190h) [reset = 0h]

TPTC1WRMPUSTADD1 is shown in Figure 4-916 and described in Table 4-959.

Return to Summary Table.

Figure 4-332. TPTC1WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD1																															
R/W-0h																															

Table 4-356. TPTC1WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC1

4.7.4.32 TPTC1WRMPUSTADD2 Register (Offset = 194h) [reset = 0h]

TPTC1WRMPUSTADD2 is shown in Figure 4-917 and described in Table 4-960.

Return to Summary Table.

Figure 4-333. TPTC1WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD2																															
R/W-0h																															

Table 4-357. TPTC1WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC1

4.7.4.33 TPTC1WRMPUSTADD3 Register (Offset = 198h) [reset = 0h]

TPTC1WRMPUSTADD3 is shown in Figure 4-918 and described in Table 4-961.

Return to Summary Table.

Figure 4-334. TPTC1WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD3																															
R/W-0h																															

Table 4-358. TPTC1WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC1

4.7.4.34 TPTC1WRMPUSTADD4 Register (Offset = 19Ch) [reset = 0h]

TPTC1WRMPUSTADD4 is shown in Figure 4-919 and described in Table 4-962.

Return to Summary Table.

Figure 4-335. TPTC1WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD4																															
R/W-0h																															

Table 4-359. TPTC1WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC1

4.7.4.35 TPTC1WRMPUSTADD5 Register (Offset = 1A0h) [reset = 0h]

TPTC1WRMPUSTADD5 is shown in Figure 4-920 and described in Table 4-963.

Return to Summary Table.

Figure 4-336. TPTC1WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD5																															
R/W-0h																															

Table 4-360. TPTC1WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC1

4.7.4.36 TPTC1WRMPUENDADD0 Register (Offset = 1ACh) [reset = 0h]

TPTC1WRMPUENDADD0 is shown in Figure 4-921 and described in Table 4-964.

Return to Summary Table.

Figure 4-337. TPTC1WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD0																															
R/W-0h																															

Table 4-361. TPTC1WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC1

4.7.4.37 TPTC1WRMPUENDADD1 Register (Offset = 1B0h) [reset = 0h]

TPTC1WRMPUENDADD1 is shown in Figure 4-922 and described in Table 4-965.

Return to Summary Table.

Figure 4-338. TPTC1WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD1																															
R/W-0h																															

Table 4-362. TPTC1WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC1

4.7.4.38 TPTC1WRMPUENDADD2 Register (Offset = 1B4h) [reset = 0h]

TPTC1WRMPUENDADD2 is shown in Figure 4-923 and described in Table 4-966.

Return to Summary Table.

Figure 4-339. TPTC1WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD2																															
R/W-0h																															

Table 4-363. TPTC1WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC1

4.7.4.39 TPTC1WRMPUENDADD3 Register (Offset = 1B8h) [reset = 0h]

TPTC1WRMPUENDADD3 is shown in Figure 4-924 and described in Table 4-967.

Return to Summary Table.

Figure 4-340. TPTC1WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD3																															
R/W-0h																															

Table 4-364. TPTC1WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC1

4.7.4.40 TPTC1WRMPUENDADD4 Register (Offset = 1BCh) [reset = 0h]

TPTC1WRMPUENDADD4 is shown in Figure 4-925 and described in Table 4-968.

Return to Summary Table.

Figure 4-341. TPTC1WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD4																															
R/W-0h																															

Table 4-365. TPTC1WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC1

4.7.4.41 TPTC1WRMPUENDADD5 Register (Offset = 1C0h) [reset = 0h]

TPTC1WRMPUENDADD5 is shown in Figure 4-926 and described in Table 4-969.

Return to Summary Table.

Figure 4-342. TPTC1WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD5																															
R/W-0h																															

Table 4-366. TPTC1WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC1

4.7.4.42 TPTC1WRMPUERRADD Register (Offset = 1CCh) [reset = 0h]

TPTC1WRMPUERRADD is shown in Figure 4-927 and described in Table 4-970.

Return to Summary Table.

Figure 4-343. TPTC1WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUERRADD																															
R-0h																															

Table 4-367. TPTC1WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC1

4.7.4.43 TPTC1RDMPUSTADD0 Register (Offset = 1D0h) [reset = 0h]

TPTC1RDMPUSTADD0 is shown in Figure 4-928 and described in Table 4-971.

Return to Summary Table.

Figure 4-344. TPTC1RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD0																															
R/W-0h																															

Table 4-368. TPTC1RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC1

4.7.4.44 TPTC1RDMPUSTADD1 Register (Offset = 1D4h) [reset = 0h]

TPTC1RDMPUSTADD1 is shown in Figure 4-929 and described in Table 4-972.

Return to Summary Table.

Figure 4-345. TPTC1RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD1																															
R/W-0h																															

Table 4-369. TPTC1RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC1

4.7.4.45 TPTC1RDMPUSTADD2 Register (Offset = 1D8h) [reset = 0h]

TPTC1RDMPUSTADD2 is shown in Figure 4-930 and described in Table 4-973.

Return to Summary Table.

Figure 4-346. TPTC1RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD2																															
R/W-0h																															

Table 4-370. TPTC1RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC1

4.7.4.46 TPTC1RDMPUSTADD3 Register (Offset = 1DCh) [reset = 0h]

TPTC1RDMPUSTADD3 is shown in Figure 4-931 and described in Table 4-974.

Return to Summary Table.

Figure 4-347. TPTC1RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD3																															
R/W-0h																															

Table 4-371. TPTC1RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC1

4.7.4.47 TPTC1RDMPUSTADD4 Register (Offset = 1E0h) [reset = 0h]

TPTC1RDMPUSTADD4 is shown in Figure 4-932 and described in Table 4-975.

Return to Summary Table.

Figure 4-348. TPTC1RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD4																															
R/W-0h																															

Table 4-372. TPTC1RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC1

4.7.4.48 TPTC1RDMPUSTADD5 Register (Offset = 1E4h) [reset = 0h]

TPTC1RDMPUSTADD5 is shown in Figure 4-933 and described in Table 4-976.

Return to Summary Table.

Figure 4-349. TPTC1RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD5																															
R/W-0h																															

Table 4-373. TPTC1RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC1

4.7.4.49 TPTC1RDMPUENDADD0 Register (Offset = 1F0h) [reset = 0h]

TPTC1RDMPUENDADD0 is shown in Figure 4-934 and described in Table 4-977.

Return to Summary Table.

Figure 4-350. TPTC1RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD0																															
R/W-0h																															

Table 4-374. TPTC1RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC1

4.7.4.50 TPTC1RDMPUENDADD1 Register (Offset = 1F4h) [reset = 0h]

TPTC1RDMPUENDADD1 is shown in Figure 4-935 and described in Table 4-978.

Return to Summary Table.

Figure 4-351. TPTC1RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD1																															
R/W-0h																															

Table 4-375. TPTC1RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC1

4.7.4.51 TPTC1RDMPUENDADD2 Register (Offset = 1F8h) [reset = 0h]

TPTC1RDMPUENDADD2 is shown in Figure 4-936 and described in Table 4-979.

Return to Summary Table.

Figure 4-352. TPTC1RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD2																															
R/W-0h																															

Table 4-376. TPTC1RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC1

4.7.4.52 TPTC1RDMPUENDADD3 Register (Offset = 1FCh) [reset = 0h]

TPTC1RDMPUENDADD3 is shown in Figure 4-937 and described in Table 4-980.

Return to Summary Table.

Figure 4-353. TPTC1RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD3																															
R/W-0h																															

Table 4-377. TPTC1RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC1

4.7.4.53 TPTC1RDMPUENDADD4 Register (Offset = 200h) [reset = 0h]

TPTC1RDMPUENDADD4 is shown in Figure 4-938 and described in Table 4-981.

Return to Summary Table.

Figure 4-354. TPTC1RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD4																															
R/W-0h																															

Table 4-378. TPTC1RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC1

4.7.4.54 TPTC1RDMPUENDADD5 Register (Offset = 204h) [reset = 0h]

TPTC1RDMPUENDADD5 is shown in Figure 4-939 and described in Table 4-982.

Return to Summary Table.

Figure 4-355. TPTC1RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD5																															
R/W-0h																															

Table 4-379. TPTC1RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC1

4.7.4.55 TPTC1RDMPUERRADD Register (Offset = 210h) [reset = 0h]

TPTC1RDMPUERRADD is shown in Figure 4-940 and described in Table 4-983.

Return to Summary Table.

Figure 4-356. TPTC1RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUERRADD																															
R-0h																															

Table 4-380. TPTC1RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC1

4.7.4.56 TPTCMPUVALIDCFG Register (Offset = 214h) [reset = 0h]

TPTCMPUVALIDCFG is shown in Figure 4-941 and described in Table 4-984.

Return to Summary Table.

Figure 4-357. TPTCMPUVALIDCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPTC1RDMPURNGVLD								TPTC1WRMPURNGVLD							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPURNGVLD								TPTC0WRMPURNGVLD							
R/W-0h								R/W-0h							

Table 4-381. TPTCMPUVALIDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TPTC1RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC1. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
23-16	TPTC1WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC1. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
15-8	TPTC0RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC0. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
7-0	TPTC0WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC0. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled

4.7.4.57 TPTCMPUENCFG Register (Offset = 218h) [reset = 0h]

TPTCMPUENCFG is shown in Figure 4-942 and described in Table 4-985.

Return to Summary Table.

Figure 4-358. TPTCMPUENCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPTC1RDMPUERRCLR	TPTC1WRMPUERRCLR	TPTC0RDMPUERRCLR	TPTC0WRMPUERRCLR	TPTC1RDMPUEN	TPTC1WRMPUEN	TPTC0RDMPUEN	TPTC0WRMPUEN
0h	0h	0h	0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-382. TPTCMPUENCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPTC1RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC1. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
6	TPTC1WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC1. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
5	TPTC0RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC0. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
4	TPTC0WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC0. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
3	TPTC1RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC1. 0 : MPU is disabled 1 : MPU is enabled
2	TPTC1WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC1. 0 : MPU is disabled 1 : MPU is enabled
1	TPTC0RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC0. 0 : MPU is disabled 1 : MPU is enabled
0	TPTC0WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC0. 0 : MPU is disabled 1 : MPU is enabled

4.7.4.58 TESTPATTERNRX1ICFG Register (Offset = 21Ch) [reset = 00010000h]

TESTPATTERNRX1ICFG is shown in Figure 4-943 and described in Table 4-986.

Return to Summary Table.

Figure 4-359. TESTPATTERNRX1ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1IINCR																TSTPATRX1IOFFSET															
R/W-1h																R/W-0h															

Table 4-383. TESTPATTERNRX1ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.7.4.59 TESTPATTERNRX2ICFG Register (Offset = 220h) [reset = 00010000h]

TESTPATTERNRX2ICFG is shown in Figure 4-944 and described in Table 4-987.

Return to Summary Table.

Figure 4-360. TESTPATTERNRX2ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2IINCR																TSTPATRX2IOFFSET															
R/W-1h																R/W-0h															

Table 4-384. TESTPATTERNRX2ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 1.
15-0	TSTPATRX2IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 1.

4.7.4.60 TESTPATTERNRX3ICFG Register (Offset = 224h) [reset = 00010000h]

TESTPATTERNRX3ICFG is shown in Figure 4-945 and described in Table 4-988.

Return to Summary Table.

Figure 4-361. TESTPATTERNRX3ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3IINCR																TSTPATRX3IOFFSET															
R/W-1h																R/W-0h															

Table 4-385. TESTPATTERNRX3ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 2
15-0	TSTPATRX3IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 2

4.7.4.61 TESTPATTERNRX4ICFG Register (Offset = 228h) [reset = 00010000h]

TESTPATTERNRX4ICFG is shown in Figure 4-946 and described in Table 4-989.

Return to Summary Table.

Figure 4-362. TESTPATTERNRX4ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4IINCR																TSTPATRX4IOFFSET															
R/W-1h																R/W-0h															

Table 4-386. TESTPATTERNRX4ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 3
15-0	TSTPATRX4IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 3

4.7.4.62 TESTPATTERNRX1QCFG Register (Offset = 22Ch) [reset = 00010000h]

TESTPATTERNRX1QCFG is shown in Figure 4-947 and described in Table 4-990.

Return to Summary Table.

Figure 4-363. TESTPATTERNRX1QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1QINCR																TSTPATRX1QOFFSET															
R/W-1h																R/W-0h															

Table 4-387. TESTPATTERNRX1QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.7.4.63 TESTPATTERNRX2QCFG Register (Offset = 230h) [reset = 00010000h]

TESTPATTERNRX2QCFG is shown in Figure 4-948 and described in Table 4-991.

Return to Summary Table.

Figure 4-364. TESTPATTERNRX2QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2QINCR																TSTPATRX2QOFFSET															
R/W-1h																R/W-0h															

Table 4-388. TESTPATTERNRX2QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 1.
15-0	TSTPATRX2QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 1.

4.7.4.64 TESTPATTERNRX3QCFG Register (Offset = 234h) [reset = 00010000h]

TESTPATTERNRX3QCFG is shown in Figure 4-949 and described in Table 4-992.

Return to Summary Table.

Figure 4-365. TESTPATTERNRX3QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3QINCR																TSTPATRX3QOFFSET															
R/W-1h																R/W-0h															

Table 4-389. TESTPATTERNRX3QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 2
15-0	TSTPATRX3QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 2

4.7.4.65 TESTPATTERNRX4QCFG Register (Offset = 238h) [reset = 00010000h]

TESTPATTERNRX4QCFG is shown in Figure 4-950 and described in Table 4-993.

Return to Summary Table.

Figure 4-366. TESTPATTERNRX4QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4QINCR																TSTPATRX4QOFFSET															
R/W-1h																R/W-0h															

Table 4-390. TESTPATTERNRX4QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 3
15-0	TSTPATRX4QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 3

4.7.4.66 TESTPATTERNVLDCFG Register (Offset = 23Ch) [reset = 8h]

TESTPATTERNVLDCFG is shown in Figure 4-951 and described in Table 4-994.

Return to Summary Table.

Figure 4-367. TESTPATTERNVLDCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU					TSTPATGENEN		
R-0h					R/W-0h		
7	6	5	4	3	2	1	0
TSTPATVLDCNT							
R/W-8h							

Table 4-391. TESTPATTERNVLDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	
10-8	TSTPATGENEN	R/W	0h	Enable for test pattern generator. This is used to Mux with the functional data from BSS. 000 -->Disable, 111-->Enable, Others are reserved.
7-0	TSTPATVLDCNT	R/W	8h	Number of DSS Interconnect clocks (200 MHz) between successive samples for the test pattern gen.

4.7.4.67 TPCC1PARSTATCFG Register (Offset = 258h) [reset = 0h]

TPCC1PARSTATCFG is shown in Figure 4-952 and described in Table 4-995.

Return to Summary Table.

Figure 4-368. TPCC1PARSTATCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU				TPCC1PARITYTSTEN	TPCC1PARITYEN	TPCC1PARITYCLR	TPCC1PARITYSTAT
R-0h				R/W-0h	R/W-0h	0h	R-0h
7	6	5	4	3	2	1	0
TPCC1PARITYSTAT							
R-0h							

Table 4-392. TPCC1PARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11	TPCC1PARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in TPCC
10	TPCC1PARITYEN	R/W	0h	Enable bit for the Parity computation in TPCC
9	TPCC1PARITYCLR		0h	Clear bit for the Parity error from TPCC Write 0x1 to clear the status. It is a wspecial access type; a write to this field generates a pulse.
8-0	TPCC1PARITYSTAT	R	0h	Parity address from TPCC

4.7.4.68 DMMSWINT1 Register (Offset = 260h) [reset = 0h]

DMMSWINT1 is shown in Figure 4-953 and described in Table 4-996.

Return to Summary Table.

Figure 4-369. DMMSWINT1 Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2	DMMCQWREN	DMMCQPINPONSEL	DMMCPBPMMEMSEL	DMMCPBPMWREN	DMMCPBPMPINPONSEL	DMMADCBUFWREN	DMMADCBUFPINPONSEL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
NU1							
R-0h							
7	6	5	4	3	2	1	0
NU1							
R-0h							

Table 4-393. DMMSWINT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	Not Used
22	DMMCQWREN	R/W	0h	CQ Write Enable from DMM. 0 --> Write to CQ memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from CQ_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMCQPINPONSEL register.
21	DMMCQPINPONSEL	R/W	0h	CQ Ping Pong select for HIL Mode
20	DMMCPBPMMEMSEL	R/W	0h	Select signal for Muxing between HW Registers/Memory for CPBPM data. 0 --> Read access from CPBPM_MEM slave in DSS interconnect will be routed to HW Registers which is populated by DFE/RampGen, 1 --> Read access from CPBPM_MEM slave in DSS interconnect will be routed to appropriate CPBPM memory (Ping/Pong).
19	DMMCPBPMWREN	R/W	0h	CPBPM Write Enable from DMM. 0 --> Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Ping-pong select will come from DMMCPBPMPINPONSEL register.
18	DMMCPBPMPINPONSEL	R/W	0h	CP BPM Ping Pong select for HIL Mode
17	DMMADCBUFWREN	R/W	0h	ADC Buffer Write Enable from DMM. 0 --> Write to ADC BUF memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from ADCBUF_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMADCBUFPINPONSEL register.
16	DMMADCBUFPINPONSEL	R/W	0h	ADC Buffer Ping Pong select for HIL Mode
15-0	NU1	R	0h	Not Used

4.7.4.69 DSSINTRCFG Register (Offset = 270h) [reset = 0h]

DSSINTRCFG is shown in Figure 4-954 and described in Table 4-997.

Return to Summary Table.

Figure 4-370. DSSINTRCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
LGFRAMESTRTINTMUXSEL		PINPONINTMUXSEL		CHIRPAVLINTMUXSEL		FRAMESTRTINTMUXSEL	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 4-394. DSSINTRCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Not Used
7-6	LGFRAMESTRTINTMUXSEL	R/W	0h	[0] - Used to select between DFE and DMM Global Logical Frame Start CFG Bit. 0 --> Select DFE Logical Frame Start, 1--> Select DMM Global Logical Frame start CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 3.
5-4	PINPONINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Ping Pong Switch, 1--> Select DMM Global Ping Pong Switch CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 2.
3-2	CHIRPAVLINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Chirp Available, 1--> Select DMM Global Chirp Available CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 1.
1-0	FRAMESTRTINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Frame Start, 1--> Select DMM Global Frame start CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 0.

4.7.4.70 MPUMSTIDCFG1 Register (Offset = 274h) [reset = 1A191514h]

MPUMSTIDCFG1 is shown in Figure 4-955 and described in Table 4-998.

Return to Summary Table.

Figure 4-371. MPUMSTIDCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUMSTID3								MPUMSTID2							
R/W-1Ah								R/W-19h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPUMSTID1								MPUMSTID0							
R/W-15h								R/W-14h							

Table 4-395. MPUMSTIDCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MPUMSTID3	R/W	1Ah	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID3. Defaule value maps to RS232 Port
23-16	MPUMSTID2	R/W	19h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID2. Defaule value maps to MSS DAP Port
15-8	MPUMSTID1	R/W	15h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID0. Defaule value maps to MSS CR4 Read Port
7-0	MPUMSTID0	R/W	14h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID0. Defaule value maps to MSS CR4 Read Port

4.7.4.71 MPUMSTIDCFG2 Register (Offset = 278h) [reset = 1A191514h]

MPUMSTIDCFG2 is shown in Figure 4-956 and described in Table 4-999.

Return to Summary Table.

Figure 4-372. MPUMSTIDCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUMSTID7								MPUMSTID6							
R/W-1Ah								R/W-19h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPUMSTID5								MPUMSTID4							
R/W-15h								R/W-14h							

Table 4-396. MPUMSTIDCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MPUMSTID7	R/W	1Ah	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID7. Defaule value maps to RS232 Port
23-16	MPUMSTID6	R/W	19h	Allowed MSTID6. Defaule value maps to MSS DAP Port
15-8	MPUMSTID5	R/W	15h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID5 Defaule value maps to MSS CR4 Write Port
7-0	MPUMSTID4	R/W	14h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID4. Defaule value maps to MSS CR4 Read Port

4.7.4.72 MPUMSTIDCFG3 Register (Offset = 27Ch) [reset = FFh]

MPUMSTIDCFG3 is shown in Figure 4-957 and described in Table 4-1000.

Return to Summary Table.

Figure 4-373. MPUMSTIDCFG3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				MPUMSTIDEN	RESERVED	MPUERRCLR	RESERVED
R/W-0h				R/W-0h	R/W-0h	0h	0h
15	14	13	12	11	10	9	8
MPUERRMSTID							
R-0h							
7	6	5	4	3	2	1	0
MPUMSTIDVLD							
R/W-FFh							

Table 4-397. MPUMSTIDCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	MPUMSTIDEN	R/W	0h	Enable control for Master ID based MPU 0 --> Disabled, 1 --> Enabled
18	RESERVED	R/W	0h	Reserved
17	MPUERRCLR		0h	Error clear pulse for Master ID based MPU Write 0x1 to clear the previous error status: it is a wspecial access type; a write to this field generates a pulse.
16	RESERVED		0h	Reserved
15-8	MPUERRMSTID	R	0h	Error status field. Provides the Master ID which is not part of the allowed list which caused an error.
7-0	MPUMSTIDVLD	R/W	FFh	Master ID valid. Each bit corresponds to the MPUMSTID[7:0] 0 : Master ID entry is valid 1 : Master ID entry is not valid and entry does not have access to DSS CFG Space

4.7.4.73 HSRAM1ECCCFG Register (Offset = 280h) [reset = 0h]

HSRAM1ECCCFG is shown in Figure 4-958 and described in Table 4-1001.

Return to Summary Table.

Figure 4-374. HSRAM1ECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	HSRAM1ECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
HSRAM1ECCREPAIREDBIT	HSRAM1ECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
HSRAM1ECCFAULTADDRESS				HSRAM1ECCERRCLR	HSRAM1ECCEN	HSRAM1ECCINITDONE	HSRAM1ECCINIT
R-0h				0h	R/W-0h	R-0h	0h

Table 4-398. HSRAM1ECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	HSRAM1ECCREPAIREDBIT	R	0h	Bit position of the repaired bit in HSRAM1
14-4	HSRAM1ECCFAULTADDRESS	R	0h	ECC Fault address in HSRAM1
3	HSRAM1ECCERRCLR		0h	Clear bit for ECC Error Indication in HSRAM1 Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	HSRAM1ECCEN	R/W	0h	Enable for ECC in HSRAM1
1	HSRAM1ECCINITDONE	R	0h	Done status for ECC Init for HSRAM1
0	HSRAM1ECCINIT		0h	ECC Init For HSRAM1: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.74 DATATRRAMECCCFG Register (Offset = 288h) [reset = 0h]

DATATRRAMECCCFG is shown in Figure 4-959 and described in Table 4-1002.

Return to Summary Table.

Figure 4-375. DATATRRAMECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU			DATATRRAMECCREPAIREDBIT				
R/W-0h			R-0h				
15	14	13	12	11	10	9	8
DATATRRAMECCREPAIREDBIT			DATATRRAMECCFAULTADDRESS				
R-0h			R-0h				
7	6	5	4	3	2	1	0
DATATRRAMECCFAULTADDRESS				DATATRRAMECCERRCLR	DATATRRAMECCEN	DATATRRAMECCINITDONE	DATATRRAMECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-399. DATATRRAMECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU	R	0h	Not Used
20-13	DATATRRAMECCREPAIREDBIT	R	0h	Bit position of the repaired bit in DATATRRAM
12-4	DATATRRAMECCFAULTADDRESS	R	0h	ECC Fault address in DATATRRAM
3	DATATRRAMECCERRCLR	W	0h	Clear bit for ECC Error Indication in DATATRRAM Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	DATATRRAMECCEN	R/W	0h	Enable for ECC in DATATRRAM
1	DATATRRAMECCINITDONE	R	0h	Done status for ECC Init for Data Transfer RAM
0	DATATRRAMECCINIT	W	0h	ECC Init For Data Transfer RAM: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.75 ADCBUFPINGECCCFG Register (Offset = 28Ch) [reset = 0h]

ADCBUFPINGECCCFG is shown in Figure 4-960 and described in Table 4-1003.

Return to Summary Table.

Figure 4-376. ADCBUFPINGECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	ADCBUFPINGECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
ADCBUFPINGECCREPAIREDBIT	ADCBUFPINGECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
ADCBUFPINGECCFAULTADDRESS				ADCBUFPINGECCERRCLR	ADCBUFPINGECCEN	ADCBUFPINGECCINITDONE	ADCBUFPINGECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-400. ADCBUFPINGECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	ADCBUFPINGECCREPAIREDBIT	R	0h	Bit position of the repaired bit in ADC Buffer Ping Memory
14-4	ADCBUFPINGECCFAULTADDRESS	R	0h	ECC Fault address in ADC Buffer Ping Memory
3	ADCBUFPINGECCERRCLR	W	0h	Clear bit for ECC Error Indication in ADC Buffer Ping Memory Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	ADCBUFPINGECCEN	R/W	0h	Enable for ECC in ADC Buffer Ping Memory
1	ADCBUFPINGECCINITDONE	R	0h	Done status for ECC Init for ADC Buffer Ping Memory
0	ADCBUFPINGECCINIT	W	0h	ECC Init For ADC Buffer Ping Memory: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.76 ADCBUFPONGECCCFG Register (Offset = 290h) [reset = 0h]

ADCBUFPONGECCCFG is shown in Figure 4-961 and described in Table 4-1004.

Return to Summary Table.

Figure 4-377. ADCBUFPONGECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	ADCBUFPONGECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
ADCBUFPONGECCREPAIREDBIT	ADCBUFPONGECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
ADCBUFPONGECCFAULTADDRESS				ADCBUFPONGECCERRCLR	ADCBUFPONGECCEN	ADCBUFPONGECCINITDONE	ADCBUFPONGECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-401. ADCBUFPONGECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	ADCBUFPONGECCREPAIREDBIT	R	0h	Bit position of the repaired bit in ADC Buffer Pong Memory
14-4	ADCBUFPONGECCFAULTADDRESS	R	0h	ECC Fault address in ADC Buffer Pong Memory
3	ADCBUFPONGECCERRCLR	W	0h	Clear bit for ECC Error Indication in ADC Buffer Pong Memory Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	ADCBUFPONGECCEN	R/W	0h	Enable for ECC in ADC Buffer Pong Memory
1	ADCBUFPONGECCINITDONE	R	0h	Done status for ECC Init for ADC Buffer Pong Memory
0	ADCBUFPONGECCINIT	W	0h	ECC Init For ADC Buffer Pong Memory: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.77 UMAP0PARITYCFG1 Register (Offset = 29Ch) [reset = 0h]

UMAP0PARITYCFG1 is shown in Figure 4-962 and described in Table 4-1005.

Return to Summary Table.

Figure 4-378. UMAP0PARITYCFG1 Register

31	30	29	28	27	26	25	24
NU						UMAP0BANK23ADDOUT	
R/W-0h						R-0h	
23	22	21	20	19	18	17	16
UMAP0BANK23ADDOUT							
R-0h							
15	14	13	12	11	10	9	8
UMAP0BANK23ADDOUT	UMAP0BANK01ADDOUT						
R-0h	R-0h						
7	6	5	4	3	2	1	0
UMAP0BANK01ADDOUT				UMAP0BANK23ERROUT	UMAP0BANK01ERROUT	UMAP0PARERRCLR	UMAP0PAREN
R-0h				R-0h	R-0h	0h	R/W-0h

Table 4-402. UMAP0PARITYCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU	R/W	0h	Not Used
25-15	UMAP0BANK23ADDOUT	R	0h	Address corresponding to the parity error in Bank2 or Bank3 of UMAP0.
14-4	UMAP0BANK01ADDOUT	R	0h	Address corresponding to the parity error in Bank0 or Bank1 of UMAP0
3	UMAP0BANK23ERROUT	R	0h	Parity Error indication from either Bank2 or Bank3 of UMAP0
2	UMAP0BANK01ERROUT	R	0h	Parity Error indication from either Bank 0 or Bank1 of UMAP0
1	UMAP0PARERRCLR		0h	Clear pulse for all the error status from UMAP0 parity check logic. Self-clearing pulse Write 0x1 to clear the error status
0	UMAP0PAREN	R/W	0h	Enable for UMAP0 Parity Check logic. Assumed to be static. 0 --> Disable 1 --> Enable

4.7.4.78 UMAP0PARITYCFG2 Register (Offset = 2A0h) [reset = 0h]

UMAP0PARITYCFG2 is shown in Figure 4-963 and described in Table 4-1006.

Return to Summary Table.

Figure 4-379. UMAP0PARITYCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP0BANK1BITOUT																UMAP0BANK0BITOUT															
R-0h																R-0h															

Table 4-403. UMAP0PARITYCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP0BANK1BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank1.
15-0	UMAP0BANK0BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank0.

4.7.4.79 UMAP0PARITYCFG3 Register (Offset = 2A4h) [reset = 0h]

UMAP0PARITYCFG3 is shown in Figure 4-964 and described in Table 4-1007.

Return to Summary Table.

Figure 4-380. UMAP0PARITYCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP0BANK3BITOUT																UMAP0BANK2BITOUT															
R-0h																R-0h															

Table 4-404. UMAP0PARITYCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP0BANK3BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank3.
15-0	UMAP0BANK2BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank2.

4.7.4.80 UMAP1PARITYCFG1 Register (Offset = 2A8h) [reset = 0h]

UMAP1PARITYCFG1 is shown in Figure 4-965 and described in Table 4-1008.

Return to Summary Table.

Figure 4-381. UMAP1PARITYCFG1 Register

31	30	29	28	27	26	25	24
NU						UMAP1BANK23ADDOUT	
R-0h						R-0h	
23	22	21	20	19	18	17	16
UMAP1BANK23ADDOUT							
R-0h							
15	14	13	12	11	10	9	8
UMAP1BANK23ADDOUT	UMAP1BANK01ADDOUT						
R-0h	R-0h						
7	6	5	4	3	2	1	0
UMAP1BANK01ADDOUT				UMAP1BANK23ERROUT	UMAP1BANK01ERROUT	UMAP1PARERRCLR	UMAP1PAREN
R-0h				R-0h	R-0h	W-0h	R/W-0h

Table 4-405. UMAP1PARITYCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU	R	0h	Not Used
25-15	UMAP1BANK23ADDOUT	R	0h	Address corresponding to the parity error in Bank2 or Bank3 of UMAP1.
14-4	UMAP1BANK01ADDOUT	R	0h	Address corresponding to the parity error in Bank0 or Bank1 of UMAP1
3	UMAP1BANK23ERROUT	R	0h	Parity Error indication from either Bank2 or Bank3 of UMAP1
2	UMAP1BANK01ERROUT	R	0h	Parity Error indication from either Bank 0 or Bank1 of UMAP1
1	UMAP1PARERRCLR	W	0h	Clear pulse for all the error status from UMAP1 parity check logic. Self-clearing pulse Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP1PAREN	R/W	0h	Enable for UMAP1 Parity Check logic. Assumed to be static. 0 --> Disable 1 --> Enable

4.7.4.81 UMAP1PARITYCFG2 Register (Offset = 2ACh) [reset = 0h]

UMAP1PARITYCFG2 is shown in Figure 4-966 and described in Table 4-1009.

Return to Summary Table.

Figure 4-382. UMAP1PARITYCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP1BANK1BITOUT																UMAP1BANK0BITOUT															
R-0h																R-0h															

Table 4-406. UMAP1PARITYCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP1BANK1BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank1.
15-0	UMAP1BANK0BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank0.

4.7.4.82 UMAP1PARITYCFG3 Register (Offset = 2B0h) [reset = 0h]

UMAP1PARITYCFG3 is shown in Figure 4-967 and described in Table 4-1010.

Return to Summary Table.

Figure 4-383. UMAP1PARITYCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP1BANK3BITOUT																UMAP1BANK2BITOUT															
R-0h																R-0h															

Table 4-407. UMAP1PARITYCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP1BANK3BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank3.
15-0	UMAP1BANK2BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank2.

4.7.4.83 ESMGRP2MASKCFG Register (Offset = 2B4h) [reset = FFFFFFFFh]

ESMGRP2MASKCFG is shown in Figure 4-968 and described in Table 4-1011.

Return to Summary Table.

Figure 4-384. ESMGRP2MASKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGRP2MASK																															
R/W-FFFFFFFFh																															

Table 4-408. ESMGRP2MASKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGRP2MASK	R/W	FFFFFFFFh	Bbit level mask for each of the error signal connected to ESM Group2 input.

4.7.4.84 L2MEMINITCFG1 Register (Offset = 2B8h) [reset = 0h]

L2MEMINITCFG1 is shown in Figure 4-969 and described in Table 4-1012.

Return to Summary Table.

Figure 4-385. L2MEMINITCFG1 Register

31	30	29	28	27	26	25	24
UMAP1BANK3 PARINITDONE	UMAP1BANK2 PARINITDONE	UMAP1BANK1 PARINITDONE	UMAP1BANK0 PARINITDONE	UMAP0BANK3 PARINITDONE	UMAP0BANK2 PARINITDONE	UMAP0BANK1 PARINITDONE	UMAP0BANK0 PARINITDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
UMAP1BANK3 DATAINITDONE	UMAP1BANK2 DATAINITDONE	UMAP1BANK1 DATAINITDONE	UMAP1BANK0 DATAINITDONE	UMAP0BANK3 DATAINITDONE	UMAP0BANK2 DATAINITDONE	UMAP0BANK1 DATAINITDONE	UMAP0BANK0 DATAINITDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
UMAP1BANK3 PARINIT	UMAP1BANK2 PARINIT	UMAP1BANK1 PARINIT	UMAP1BANK0 PARINIT	UMAP0BANK3 PARINIT	UMAP0BANK2 PARINIT	UMAP0BANK1 PARINIT	UMAP0BANK0 PARINIT
0h	0h	0h	0h	0h	0h	0h	0h
7	6	5	4	3	2	1	0
UMAP1BANK3 DATAINIT	UMAP1BANK2 DATAINIT	UMAP1BANK1 DATAINIT	UMAP1BANK0 DATAINIT	UMAP0BANK3 DATAINIT	UMAP0BANK2 DATAINIT	UMAP0BANK1 DATAINIT	UMAP0BANK0 DATAINIT
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-409. L2MEMINITCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	UMAP1BANK3PARINITDONE	R	0h	Init Done status from UMAP1 Bank3 Parity memory
30	UMAP1BANK2PARINITDONE	R	0h	Init Done status from UMAP1 Bank2 Parity memory
29	UMAP1BANK1PARINITDONE	R	0h	Init Done status from UMAP1 Bank1 Parity memory
28	UMAP1BANK0PARINITDONE	R	0h	Init Done status from UMAP1 Bank0 Parity memory
27	UMAP0BANK3PARINITDONE	R	0h	Init Done status from UMAP0 Bank3 Parity memory
26	UMAP0BANK2PARINITDONE	R	0h	Init Done status from UMAP0 Bank2 Parity memory
25	UMAP0BANK1PARINITDONE	R	0h	Init Done status from UMAP0 Bank1 Parity memory
24	UMAP0BANK0PARINITDONE	R	0h	Init Done status from UMAP0 Bank0 Parity memory
23	UMAP1BANK3DATAINITDONE	R	0h	Init Done status from UMAP1 Bank3 Data memory
22	UMAP1BANK2DATAINITDONE	R	0h	Init Done status from UMAP1 Bank2 Data memory
21	UMAP1BANK1DATAINITDONE	R	0h	Init Done status from UMAP1 Bank1 Data memory
20	UMAP1BANK0DATAINITDONE	R	0h	Init Done status from UMAP1 Bank0 Data memory
19	UMAP0BANK3DATAINITDONE	R	0h	Init Done status from UMAP0 Bank3 Data memory
18	UMAP0BANK2DATAINITDONE	R	0h	Init Done status from UMAP0 Bank2 Data memory
17	UMAP0BANK1DATAINITDONE	R	0h	Init Done status from UMAP0 Bank1 Data memory

Table 4-409. L2MEMINITCFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	UMAP0BANK0DATAINIT DONE	R	0h	Init Done status from UMAP0 Bank0 Data memory
15	UMAP1BANK3PARINIT		0h	Init trigger for UMAP1 Bank3 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
14	UMAP1BANK2PARINIT		0h	Init trigger for UMAP1 Bank2 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
13	UMAP1BANK1PARINIT		0h	Init trigger for UMAP1 Bank1 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
12	UMAP1BANK0PARINIT		0h	Init trigger for UMAP1 Bank0 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
11	UMAP0BANK3PARINIT		0h	Init trigger for UMAP0 Bank3 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
10	UMAP0BANK2PARINIT		0h	Init trigger for UMAP0 Bank2 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
9	UMAP0BANK1PARINIT		0h	Init trigger for UMAP0 Bank1 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
8	UMAP0BANK0PARINIT		0h	Init trigger for UMAP0 Bank0 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
7	UMAP1BANK3DATAINIT		0h	Init trigger for UMAP1 Bank3 Data memory: it is a wspecial access type; a write to this field generates a pulse.
6	UMAP1BANK2DATAINIT		0h	Init trigger for UMAP1 Bank2 Data memory: it is a wspecial access type; a write to this field generates a pulse.
5	UMAP1BANK1DATAINIT		0h	Init trigger for UMAP1 Bank1 Data memory: it is a wspecial access type; a write to this field generates a pulse.
4	UMAP1BANK0DATAINIT		0h	Init trigger for UMAP1 Bank0 Data memory: it is a wspecial access type; a write to this field generates a pulse.
3	UMAP0BANK3DATAINIT		0h	Init trigger for UMAP0 Bank3 Data memory: it is a wspecial access type; a write to this field generates a pulse.
2	UMAP0BANK2DATAINIT		0h	Init trigger for UMAP0 Bank2 Data memory: it is a wspecial access type; a write to this field generates a pulse.
1	UMAP0BANK1DATAINIT		0h	Init trigger for UMAP0 Bank1 Data memory: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP0BANK0DATAINIT		0h	Init trigger for UMAP0 Bank0 Data memory: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.85 L2MEMINITCFG2 Register (Offset = 2BCh) [reset = 0h]

L2MEMINITCFG2 is shown in Figure 4-970 and described in Table 4-1013.

Return to Summary Table.

Figure 4-386. L2MEMINITCFG2 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
UMAP1BANK1 PRAMINITDONE	UMAP1BANK0 PRAMINITDONE	UMAP0BANK1 PRAMINITDONE	UMAP0BANK0 PRAMINITDONE	UMAP1BANK1 PRAMINIT	UMAP1BANK0 PRAMINIT	UMAP0BANK1 PRAMINIT	UMAP0BANK0 PRAMINIT
R-0h	R-0h	R-0h	R-0h	W-0h	W-0h	W-0h	W-0h

Table 4-410. L2MEMINITCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Not Used
7	UMAP1BANK1PRAMINIT DONE	R	0h	Init Done Status for UMAP1 Bank1 PRAM memory
6	UMAP1BANK0PRAMINIT DONE	R	0h	Init Done Status for UMAP1 Bank0 PRAM memory
5	UMAP0BANK1PRAMINIT DONE	R	0h	Init Done Status for UMAP0 Bank1 PRAM memory
4	UMAP0BANK0PRAMINIT DONE	R	0h	Init Done Status for UMAP0 Bank0 PRAM memory
3	UMAP1BANK1PRAMINIT	W	0h	Init trigger for UMAP1 Bank1 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
2	UMAP1BANK0PRAMINIT	W	0h	Init trigger for UMAP1 Bank0 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
1	UMAP0BANK1PRAMINIT	W	0h	Init trigger for UMAP0 Bank1 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP0BANK0PRAMINIT	W	0h	Init trigger for UMAP0 Bank0 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.86 GEMRSTCAUSE Register (Offset = 2C0h) [reset = 00010101h]

GEMRSTCAUSE is shown in Figure 4-971 and described in Table 4-1014.

Return to Summary Table.

Figure 4-387. GEMRSTCAUSE Register

31	30	29	28	27	26	25	24
NU2							GEMRSTCAUSECLR
R-0h							0h
23	22	21	20	19	18	17	16
GEMPORCAUSE							
R-1h							
15	14	13	12	11	10	9	8
GEMGRSTCAUSE							
R-1h							
7	6	5	4	3	2	1	0
GEMLRSTCAUSE							
R-1h							

Table 4-411. GEMRSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU2	R	0h	
24	GEMRSTCAUSECLR		0h	Write 0x1 to clear the reset cause register for any previous resets: it is a wspecial access type; a write to this field generates a pulse.
23-16	GEMPORCAUSE	R	1h	DSP POR reset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMPORZ Bit 3 : Reset from Power FSM Bit 4 : Reset from STC FSM
15-8	GEMGRSTCAUSE	R	1h	DSP Greset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMGRSTN Bit 3 : Reset from Power FSM Bit 4 : Reset from STC FSM
7-0	GEMLRSTCAUSE	R	1h	DSP Lreset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMLRSTN Bit 3 : Reset from Debugss Bit 4 : Reset from Power FSM Bit 5 : Reset from STC FSM

4.7.4.87 GEMPWRSMCFG4 Register (Offset = 2CCh) [reset = 00060000h]

GEMPWRSMCFG4 is shown in Figure 4-972 and described in Table 4-1015.

Return to Summary Table.

Figure 4-388. GEMPWRSMCFG4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					GEMEVENTMASK	PWRSMLRSTHALT	PWRSMSLEEPTRIG
R-0h					R/W-1h	R/W-1h	0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 4-412. GEMPWRSMCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18	GEMEVENTMASK	R/W	1h	Mask bit for events going to DSP. When this bit is set (during GEM sleep/power down mode), the events are monitored outside and will be available for DSP to read and clear them once GEM wakes-up. The monitored events can be read from PWRSMEVNTMONSTATx registers.
17	PWRSMLRSTHALT	R/W	1h	Signal to halt DSP Power cycle state machine before de-asserting LRST of DSP. This is used during code download for the first time power up.
16	PWRSMSLEEPTRIG		0h	Sleep mode trigger for DSP power down state machine. This is honoured only when DSP is in GEM_ON state: it is a wspecial access type; a write to this field generates a pulse.
15-0	RESERVED	R	0h	Reserved

4.7.4.88 PWRSMWAKEMASK0 Register (Offset = 2D4h) [reset = FFFFFFFFh]

PWRSMWAKEMASK0 is shown in Figure 4-973 and described in Table 4-1016.

Return to Summary Table.

Figure 4-389. PWRSMWAKEMASK0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK0																															
R/W-FFFFFFFFh																															

Table 4-413. PWRSMWAKEMASK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK0	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [31:0] 1 --> Masked, 0 --> Unmasked.

4.7.4.89 PWRSMWAKEMASK1 Register (Offset = 2D8h) [reset = FFFFFFFFh]

PWRSMWAKEMASK1 is shown in Figure 4-974 and described in Table 4-1017.

Return to Summary Table.

Figure 4-390. PWRSMWAKEMASK1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK1																															
R/W-FFFFFFFFh																															

Table 4-414. PWRSMWAKEMASK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK1	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [63:32] 1 --> Masked, 0 --> Unmasked.

4.7.4.90 PWRSMWAKEMASK2 Register (Offset = 2DCh) [reset = FFFFFFFFh]

PWRSMWAKEMASK2 is shown in Figure 4-975 and described in Table 4-1018.

Return to Summary Table.

Figure 4-391. PWRSMWAKEMASK2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK2																															
R/W-FFFFFFFFh																															

Table 4-415. PWRSMWAKEMASK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK2	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [95:64] 1 --> Masked, 0 --> Unmasked.

4.7.4.91 PWRSMMISEVTMASK0 Register (Offset = 2E0h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK0 is shown in Figure 4-976 and described in Table 4-1019.

Return to Summary Table.

Figure 4-392. PWRSMMISEVTMASK0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK0																															
R/W-FFFFFFFFh																															

Table 4-416. PWRSMMISEVTMASK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK0	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[31:0] 1 --> Masked, 0 --> Unmasked.

4.7.4.92 PWRSMMISEVTMASK1 Register (Offset = 2E4h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK1 is shown in Figure 4-977 and described in Table 4-1020.

Return to Summary Table.

Figure 4-393. PWRSMMISEVTMASK1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK1																															
R/W-FFFFFFFFh																															

Table 4-417. PWRSMMISEVTMASK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK1	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[63:32] 1 --> Masked, 0 --> Unmasked.

4.7.4.93 PWRSMMISEVTMASK2 Register (Offset = 2E8h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK2 is shown in Figure 4-978 and described in Table 4-1021.

Return to Summary Table.

Figure 4-394. PWRSMMISEVTMASK2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK2																															
R/W-FFFFFFFFh																															

Table 4-418. PWRSMMISEVTMASK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK2	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[95:64] 1 --> Masked, 0 --> Unmasked.

4.7.4.94 PWRSMWAKESRCSTAT0 Register (Offset = 2ECh) [reset = 0h]

PWRSMWAKESRCSTAT0 is shown in Figure 4-979 and described in Table 4-1022.

Return to Summary Table.

Figure 4-395. PWRSMWAKESRCSTAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT0																															
R-0h																															

Table 4-419. PWRSMWAKESRCSTAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT0	R	0h	Wakeup source status bits [31:0]

4.7.4.95 PWRSMWAKESRCSTAT1 Register (Offset = 2F0h) [reset = 0h]

PWRSMWAKESRCSTAT1 is shown in Figure 4-980 and described in Table 4-1023.

Return to Summary Table.

Figure 4-396. PWRSMWAKESRCSTAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT1																															
R-0h																															

Table 4-420. PWRSMWAKESRCSTAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT1	R	0h	Wakeup source status bits [63:32]

4.7.4.96 PWRSMWAKESRCSTAT2 Register (Offset = 320h) [reset = 0h]

PWRSMWAKESRCSTAT2 is shown in Figure 4-981 and described in Table 4-1024.

Return to Summary Table.

Figure 4-397. PWRSMWAKESRCSTAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT2																															
R-0h																															

Table 4-421. PWRSMWAKESRCSTAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT2	R	0h	Wakeup source status bits [95:64]

4.7.4.97 PWRSMEVNTMONSTAT0 Register (Offset = 324h) [reset = 0h]

PWRSMEVNTMONSTAT0 is shown in Figure 4-982 and described in Table 4-1025.

Return to Summary Table.

Figure 4-398. PWRSMEVNTMONSTAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT0																															
R-0h																															

Table 4-422. PWRSMEVNTMONSTAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT0	R	0h	Missed events monitor status bits [31:0]. This is monitored when the events going to DSP are masked by GEMEVENTMASK register.

4.7.4.98 PWRSMEVNTMONSTAT1 Register (Offset = 328h) [reset = 0h]

PWRSMEVNTMONSTAT1 is shown in Figure 4-983 and described in Table 4-1026.

Return to Summary Table.

Figure 4-399. PWRSMEVNTMONSTAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT1																															
R-0h																															

Table 4-423. PWRSMEVNTMONSTAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT1	R	0h	Missed events monitor status bits [63:32]

4.7.4.99 PWRSMEVNTMONSTAT2 Register (Offset = 32Ch) [reset = 0h]

PWRSMEVNTMONSTAT2 is shown in Figure 4-984 and described in Table 4-1027.

Return to Summary Table.

Figure 4-400. PWRSMEVNTMONSTAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT2																															
R-0h																															

Table 4-424. PWRSMEVNTMONSTAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT2	R	0h	Missed events monitor status bits [95:64]

4.7.4.100 PWRSMWAKESRCSTATCLR0 Register (Offset = 330h) [reset = 0h]

PWRSMWAKESRCSTATCLR0 is shown in Figure 4-985 and described in Table 4-1028.

Return to Summary Table.

Figure 4-401. PWRSMWAKESRCSTATCLR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR0																															
0h																															

Table 4-425. PWRSMWAKESRCSTATCLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR0		0h	Clear bit for wakeup source status bits [31:0]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.101 PWRSMWAKESRCSTATCLR1 Register (Offset = 334h) [reset = 0h]

PWRSMWAKESRCSTATCLR1 is shown in Figure 4-986 and described in Table 4-1029.

Return to Summary Table.

Figure 4-402. PWRSMWAKESRCSTATCLR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR1																															
0h																															

Table 4-426. PWRSMWAKESRCSTATCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR1		0h	Clear bit for wakeup source status bits [63:32]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.102 PWRSMWAKESRCSTATCLR2 Register (Offset = 338h) [reset = 0h]

PWRSMWAKESRCSTATCLR2 is shown in Figure 4-987 and described in Table 4-1030.

Return to Summary Table.

Figure 4-403. PWRSMWAKESRCSTATCLR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR2																															
0h																															

Table 4-427. PWRSMWAKESRCSTATCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR2		0h	Clear bit for wakeup source status bits [95:64]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.7.4.103 ADCBUFCFG1 Register (Offset = 33Ch) [reset = 00010000h]

ADCBUFCFG1 is shown in Figure 4-988 and described in Table 4-1031.

Return to Summary Table.

Figure 4-404. ADCBUFCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
ADCBUFCONTSTOPPL	ADCBUFCONTSTRTPL	ADCBUFCONTMODEEN	ADCBUFWRITEMODE	RESERVED		RX3EN	RX2EN
W-0h	W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RX1EN	RX0EN	ADCBUFIQSWAP	RESERVED		ADCBUFREALONLYMODE	RESERVED	
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 4-428. ADCBUFCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved
15	ADCBUFCONTSTOPPL	W	0h	Stop Pulse for Continuous mode. The data capture will stop once this register is set. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
14	ADCBUFCONTSTRTPL	W	0h	Start Pulse for Continuous mode. The data capture will start from Address 0 once this register is set. All the other configurations like Enable, Sample Count are expected to be programmed before this pulse. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
13	ADCBUFCONTMODEEN	R/W	0h	Continuous mode enable for ADC Buffer. This is set when a fixed number of samples have to be stored in Ping/Pong and not depend on Chirp time-lines (Eg: Analog Lab characterization to stream out continuous data from DFE). Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
12	ADCBUFWRITEMODE	R/W	0h	This needs to be programmed to 0x1 in 16xx 0 --> Interleaved, 1 --> Non-interleaved
11-10	RESERVED	R/W	0h	Reserved
9	RX3EN	R/W	0h	Enable for Rx3 write
8	RX2EN	R/W	0h	Enable for Rx2 write
7	RX1EN	R/W	0h	Enable for Rx1 write
6	RX0EN	R/W	0h	Enable for Rx0 write
5	ADCBUFIQSWAP	R/W	0h	0 --> I is stored in LSB and Q is stored in MSB 1 --> Q is stored in LSB and I is stored in MSB
4-3	RESERVED	R/W	0h	Reserved
2	ADCBUFREALONLYMODE	R/W	0h	0-->Complex Data mode, 1-->Real data mode
1-0	RESERVED	R/W	0h	Reserved

4.7.4.104 ADCBUFCFG2 Register (Offset = 340h) [reset = 02000000h]

ADCBUFCFG2 is shown in Figure 4-989 and described in Table 4-1032.

Return to Summary Table.

Figure 4-405. ADCBUFCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					ADCBUFADDRX1										
R-0h					R/W-200h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					ADCBUFADDRX0										
R-0h					R/W-0h										

Table 4-429. ADCBUFCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	
26-16	ADCBUFADDRX1	R/W	200h	128 bit Address offset to be added to the internal address pointer for Rx1 writes in Non-interleaved mode.
15-11	NU1	R	0h	
10-0	ADCBUFADDRX0	R/W	0h	128 bit Address offset to be added to the internal address pointer for Rx0 writes in Non-interleaved mode.

4.7.4.105 ADCBUFCFG3 Register (Offset = 344h) [reset = 06000400h]

ADCBUFCFG3 is shown in Figure 4-990 and described in Table 4-1033.

Return to Summary Table.

Figure 4-406. ADCBUFCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					ADCBUFADDRX3										
R-0h					R/W-600h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					ADCBUFADDRX2										
R-0h					R/W-400h										

Table 4-430. ADCBUFCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	
26-16	ADCBUFADDRX3	R/W	600h	128 bit Address offset to be added to the internal address pointer for Rx3 writes in Non-interleaved mode.
15-11	NU1	R	0h	
10-0	ADCBUFADDRX2	R/W	400h	128 bit Address offset to be added to the internal address pointer for Rx2 writes in Non-interleaved mode.

4.7.4.106 ADCBUFCFG4 Register (Offset = 348h) [reset = 400h]

ADCBUFCFG4 is shown in Figure 4-991 and described in Table 4-1034.

Return to Summary Table.

Figure 4-407. ADCBUFCFG4 Register

31	30	29	28	27	26	25	24
RESERVED						ADCBUFNUMCHRPPONG	
R-0h						R/W-0h	
23	22	21	20	19	18	17	16
ADCBUFNUMCHRPPONG			ADCBUFNUMCHRPPING				
R/W-0h			R/W-0h				
15	14	13	12	11	10	9	8
ADCBUFSAMPCNT							
R/W-400h							
7	6	5	4	3	2	1	0
ADCBUFSAMPCNT							
R/W-400h							

Table 4-431. ADCBUFCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-21	ADCBUFNUMCHRPPONG	R/W	0h	Number of chirps to be stored in Pong buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Ping
20-16	ADCBUFNUMCHRPPING	R/W	0h	Number of chirps to be stored in Ping buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Pong
15-0	ADCBUFSAMPCNT	R/W	400h	No of samples to store in each Ping and Pong register in continuous mode of ADC Buffer. In real only mode this refers to the number of real samples and in complex mode, this refers to number of complex samples. This refers to the number of samples per channel. This counter increments once for every new sample from DFE (as long as 1 or more channels are enabled). The max allowed value varies depending on other configurations (No of channels enabled and real/complex data). Continous mode is expected to be only used for CZ and ADC Buffer Testpattern mode

4.7.4.107 STCPBISTSMCFG1 Register (Offset = 34Ch) [reset = 18h]

STCPBISTSMCFG1 is shown in Figure 4-992 and described in Table 4-1035.

Return to Summary Table.

Figure 4-408. STCPBISTSMCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED			PBISTTESTSTATCLR	PBISTTESTSTAT		STCPBISTSMSTATE	
R-0h			W-0h	R-0h		R-0h	
15	14	13	12	11	10	9	8
STCPBISTSMSTATE				RESERVED			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED			STCPBISTCKSTPACKMASK	STCPBISTLRSTDASRTHALT	STCPBISTSMTRIG	STCPBISTEN	
R/W-0h			R/W-1h	R/W-1h	W-0h	R/W-0h	

Table 4-432. STCPBISTSMCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	Reserved
20	PBISTTESTSTATCLR	W	0h	Clear bit for PBIST Status: this is a wspecial access type - awrite to this field generates a pulse.
19-18	PBISTTESTSTAT	R	0h	PBIST status from GEM. [0] - Fail Indication [1] - Done indication
17-12	STCPBISTSMSTATE	R	0h	Current state of STC PBIST state machine
11-5	RESERVED	R/W	0h	Reserved
4	STCPBISTCKSTPACKMASK	R/W	1h	Mask bit for ignoring the clock stop ack from GEM. This will be used for ignoring clock stop ack during boot-up. 1 --> Ignore clock stop ack from GEM. 0 --> Wait for clock stop ack from GEM after giving clock stop request.
3	STCPBISTLRSTDASRTHALT	R/W	1h	Configuration to halt the state machine before the final de-assertion of LRST to enable program download. 1 --> Halt, 0 --> Proceed.
2	STCPBISTSMTRIG	W	0h	Trigger pulse for the STC PBIST state machine. This is a self-clearing pulse.
1-0	STCPBISTEN	R/W	0h	Enable for PBIST and STC. 00 - Reserved, 01 --> STC only 10 --> PBIST only 11 --> PBIST followed by STC.

4.7.4.108 STCPBISTSMCFG2 Register (Offset = 350h) [reset = 2410h]

STCPBISTSMCFG2 is shown in Figure 4-993 and described in Table 4-1036.

Return to Summary Table.

Figure 4-409. STCPBISTSMCFG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							BCK2BCKSTCEN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED		GEMPBISTROMCLKSEL		GEMTMODEVLCTASRTCNT			
R-0h		R/W-2h		R/W-10h			
7	6	5	4	3	2	1	0
GEMTMODEVLCTASRTCNT		GEMTMODEVLCTDASRTCNT					
R/W-10h		R/W-10h					

Table 4-433. STCPBISTSMCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	BCK2BCKSTCEN	R/W	0h	Enables back to Back STC.Needs to be set to 1 for self test
15-14	RESERVED	R	0h	Reserved
13-12	GEMPBISTROMCLKSEL	R/W	2h	Pbist_rom_clk_div_sel --> It is used to select the PBIST rom clock frequency . 2b00 : Div 1 (600Mhz) 2b01 : Div 2 (300Mhz) 2b10 : Div 3 (200Mhz) 2b11 : Div 4 (150Mhz)
11-6	GEMTMODEVLCTASRTCNT	R/W	10h	No of clocks (in terms of 200 MHz DSPSS Bus clock) after asserting GEM TMODE VLCT signal before proceeding to the next state. Max allowed value is 31.
5-0	GEMTMODEVLCTDASRTCNT	R/W	10h	No of clocks (in terms of 200 MHz DSPSS Bus clock) after De-asserting GEM TMODE VLCT signal before proceeding to the next state. Max allowed value is 31.

4.7.4.109 RTI2EVENTCAPTURESEL Register (Offset = 358h) [reset = 0h]

RTI2EVENTCAPTURESEL is shown in Figure 4-994 and described in Table 4-1037.

Return to Summary Table.

Figure 4-410. RTI2EVENTCAPTURESEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2									RTI2EVT1						
R-0h									R/W-0h						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1									RTI2EVT0						
R-0h									R/W-0h						

Table 4-434. RTI2EVENTCAPTURESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	
22-16	RTI2EVT1	R/W	0h	Used to Select the event to be captured for RTI2 Event1.
15-7	NU1	R	0h	
6-0	RTI2EVT0	R/W	0h	Used to Select the event to be captured for RTI2 Event0.

4.7.4.110 DSSMISC5 Register (Offset = 35Ch) [reset = 0h]

DSSMISC5 is shown in Figure 4-995 and described in Table 4-1038.

Return to Summary Table.

Figure 4-411. DSSMISC5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPCC1PARMEMINITDONE	TPCC0PARMEMINITDONE	TPCC1PARMEMINIT	TPCC0PARMEMINIT	CPBPMPIPOSELVAL	CPBPMPIPOSELCNT	CQPIPOSELVAL	CQPIPOSELCNT
R-0h	R-0h	W-0h	W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-435. DSSMISC5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPCC1PARMEMINITDONE	R	0h	Mem init done status for the TPCC1 parity memory
6	TPCC0PARMEMINITDONE	R	0h	Mem init done status for the TPCC0 parity memory
5	TPCC1PARMEMINIT	W	0h	Mem init for the TPCC1 parity memory: this is a wspecial access type, a write to this field generates a pulse.
4	TPCC0PARMEMINIT	W	0h	Mem init for the TPCC0 parity memory: this is a wspecial access type, a write to this field generates a pulse.
3	CPBPMPIPOSELVAL	R/W	0h	Ping pong select override value for CPBPM Memory. 1 --> Read access from CPBPM_MEM Slave of DSS Interconnect will be routed to ping memory and write access from CPBPM_W write will be routed to pong memory. 0 --> Read access from CPBPM_MEM Slave of DSS Interconnect will be routed to pong memory and write access from CPBPM_W write will be routed to ping memory.
2	CPBPMPIPOSELCNT	R/W	0h	Ping pong select override control for CPBPM Memory. 0 --> Ping-pong select comes from HW FSM (same as the ping-pong select for ADC Buffer)/DMMCPBPMPINPONSEL 1 --> Ping pong select for CPBPM memory is taken from SW register (CPBPMPIPOSELVAL)
1	CQPIPOSELVAL	R/W	0h	Ping pong select override value for CQ Memory. 1 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to ping memory and write access from CQ_W/DFE write will be routed to pong memory. 0 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to pong memory and write access from CQ_W/DFE write will be routed to ping memory.
0	CQPIPOSELCNT	R/W	0h	Ping pong select override control for CQ Memory. 0 --> Ping-pong select comes from HW FSM (same as the ping-pong select for ADC Buffer)/DMMCQPINPONSEL 1 --> Ping pong select for CQ memory is taken from SW register (CQPIPOSELVAL)

4.7.5 DSS_REG2 Registers

Table 4-1039 lists the memory-mapped registers for the DSS_REG2. All register offset addresses not listed in Table 4-1039 should be considered as reserved locations and the register contents should not be modified.

Table 4-436. DSS_REG2 Registers

Offset	Acronym	Register Name	Section
100h	TPTC2WRMPUSTADD0	TPTC2WRMPUSTADD0	Section 4.9.5.1
104h	TPTC2WRMPUSTADD1	TPTC2WRMPUSTADD1	Section 4.9.5.2
108h	TPTC2WRMPUSTADD2	TPTC2WRMPUSTADD2	Section 4.9.5.3
10Ch	TPTC2WRMPUSTADD3	TPTC2WRMPUSTADD3	Section 4.9.5.4
110h	TPTC2WRMPUSTADD4	TPTC2WRMPUSTADD4	Section 4.9.5.5
114h	TPTC2WRMPUSTADD5	TPTC2WRMPUSTADD5	Section 4.9.5.6
120h	TPTC2WRMPUENDADD0	TPTC2WRMPUENDADD0	Section 4.9.5.7
124h	TPTC2WRMPUENDADD1	TPTC2WRMPUENDADD1	Section 4.9.5.8
128h	TPTC2WRMPUENDADD2	TPTC2WRMPUENDADD2	Section 4.9.5.9
12Ch	TPTC2WRMPUENDADD3	TPTC2WRMPUENDADD3	Section 4.9.5.10
130h	TPTC2WRMPUENDADD4	TPTC2WRMPUENDADD4	Section 4.9.5.11
134h	TPTC2WRMPUENDADD5	TPTC2WRMPUENDADD5	Section 4.9.5.12
140h	TPTC2WRMPUERRADD	TPTC2WRMPUERRADD	Section 4.9.5.13
148h	TPTC2RDMPUSTADD0	TPTC2RDMPUSTADD0	Section 4.9.5.14
14Ch	TPTC2RDMPUSTADD1	TPTC2RDMPUSTADD1	Section 4.9.5.15
150h	TPTC2RDMPUSTADD2	TPTC2RDMPUSTADD2	Section 4.9.5.16
154h	TPTC2RDMPUSTADD3	TPTC2RDMPUSTADD3	Section 4.9.5.17
158h	TPTC2RDMPUSTADD4	TPTC2RDMPUSTADD4	Section 4.9.5.18
15Ch	TPTC2RDMPUSTADD5	TPTC2RDMPUSTADD5	Section 4.9.5.19
168h	TPTC2RDMPUENDADD0	TPTC2RDMPUENDADD0	Section 4.9.5.20
16Ch	TPTC2RDMPUENDADD1	TPTC2RDMPUENDADD1	Section 4.9.5.21
170h	TPTC2RDMPUENDADD2	TPTC2RDMPUENDADD2	Section 4.9.5.22
174h	TPTC2RDMPUENDADD3	TPTC2RDMPUENDADD3	Section 4.9.5.23
178h	TPTC2RDMPUENDADD4	TPTC2RDMPUENDADD4	Section 4.9.5.24
17Ch	TPTC2RDMPUENDADD5	TPTC2RDMPUENDADD5	Section 4.9.5.25
188h	TPTC2RDMPUERRADD	TPTC2RDMPUERRADD	Section 4.9.5.26
18Ch	TPTC3WRMPUSTADD0	TPTC3WRMPUSTADD0	Section 4.9.5.27
190h	TPTC3WRMPUSTADD1	TPTC3WRMPUSTADD1	Section 4.9.5.28
194h	TPTC3WRMPUSTADD2	TPTC3WRMPUSTADD2	Section 4.9.5.29
198h	TPTC3WRMPUSTADD3	TPTC3WRMPUSTADD3	Section 4.9.5.30
19Ch	TPTC3WRMPUSTADD4	TPTC3WRMPUSTADD4	Section 4.9.5.31
1A0h	TPTC3WRMPUSTADD5	TPTC3WRMPUSTADD5	Section 4.9.5.32
1ACh	TPTC3WRMPUENDADD0	TPTC3WRMPUENDADD0	Section 4.9.5.33
1B0h	TPTC3WRMPUENDADD1	TPTC3WRMPUENDADD1	Section 4.9.5.34
1B4h	TPTC3WRMPUENDADD2	TPTC3WRMPUENDADD2	Section 4.9.5.35
1B8h	TPTC3WRMPUENDADD3	TPTC3WRMPUENDADD3	Section 4.9.5.36
1BCh	TPTC3WRMPUENDADD4	TPTC3WRMPUENDADD4	Section 4.9.5.37
1C0h	TPTC3WRMPUENDADD5	TPTC3WRMPUENDADD5	Section 4.9.5.38
1CCh	TPTC3WRMPUERRADD	TPTC3WRMPUERRADD	Section 4.9.5.39
1D0h	TPTC3RDMPUSTADD0	TPTC3RDMPUSTADD0	Section 4.9.5.40
1D4h	TPTC3RDMPUSTADD1	TPTC3RDMPUSTADD1	Section 4.9.5.41
1D8h	TPTC3RDMPUSTADD2	TPTC3RDMPUSTADD2	Section 4.9.5.42
1DCh	TPTC3RDMPUSTADD3	TPTC3RDMPUSTADD3	Section 4.9.5.43

Table 4-436. DSS_REG2 Registers (continued)

Offset	Acronym	Register Name	Section
1E0h	TPTC3RDMPUSTADD4	TPTC3RDMPUSTADD4	Section 4.9.5.44
1E4h	TPTC3RDMPUSTADD5	TPTC3RDMPUSTADD5	Section 4.9.5.45
1F0h	TPTC3RDMPUENDADD0	TPTC3RDMPUENDADD0	Section 4.9.5.46
1F4h	TPTC3RDMPUENDADD1	TPTC3RDMPUENDADD1	Section 4.9.5.47
1F8h	TPTC3RDMPUENDADD2	TPTC3RDMPUENDADD2	Section 4.9.5.48
1FCh	TPTC3RDMPUENDADD3	TPTC3RDMPUENDADD3	Section 4.9.5.49
200h	TPTC3RDMPUENDADD4	TPTC3RDMPUENDADD4	Section 4.9.5.50
204h	TPTC3RDMPUENDADD5	TPTC3RDMPUENDADD5	Section 4.9.5.51
210h	TPTC3RDMPUERRADD	TPTC3RDMPUERRADD	Section 4.9.5.52
214h	TPTCMPUVALIDCFG2	TPTCMPUVALIDCFG2	Section 4.9.5.53
218h	TPTCMPUENCFG2	TPTCMPUENCFG2	Section 4.9.5.54
268h	L3ECCCFG1	L3ECCCFG1	Section 4.9.5.55
26Ch	L3ECCCFG2	L3ECCCFG2	Section 4.9.5.56
270h	DSS2MSSSWIRQ		Section 4.9.5.57

Complex bit access types are encoded to fit into small table cells. Table 4-1040 shows the codes that are used for access types in this section.

Table 4-437. DSS_REG2 Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.7.5.1 TPTC2WRMPUSTADD0 Register (Offset = 100h) [reset = 0h]

TPTC2WRMPUSTADD0 is shown in Figure 4-996 and described in Table 4-1041.

Return to Summary Table.

Figure 4-412. TPTC2WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD0																															
R/W-0h																															

Table 4-438. TPTC2WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC2

4.7.5.2 TPTC2WRMPUSTADD1 Register (Offset = 104h) [reset = 0h]

TPTC2WRMPUSTADD1 is shown in Figure 4-997 and described in Table 4-1042.

Return to Summary Table.

Figure 4-413. TPTC2WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD1																															
R/W-0h																															

Table 4-439. TPTC2WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC2

4.7.5.3 TPTC2WRMPUSTADD2 Register (Offset = 108h) [reset = 0h]

TPTC2WRMPUSTADD2 is shown in Figure 4-998 and described in Table 4-1043.

Return to Summary Table.

Figure 4-414. TPTC2WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD2																															
R/W-0h																															

Table 4-440. TPTC2WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC2

4.7.5.4 TPTC2WRMPUSTADD3 Register (Offset = 10Ch) [reset = 0h]

TPTC2WRMPUSTADD3 is shown in Figure 4-999 and described in Table 4-1044.

Return to Summary Table.

Figure 4-415. TPTC2WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD3																															
R/W-0h																															

Table 4-441. TPTC2WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC2

4.7.5.5 TPTC2WRMPUSTADD4 Register (Offset = 110h) [reset = 0h]

TPTC2WRMPUSTADD4 is shown in Figure 4-1000 and described in Table 4-1045.

Return to Summary Table.

Figure 4-416. TPTC2WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD4																															
R/W-0h																															

Table 4-442. TPTC2WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC2

4.7.5.6 TPTC2WRMPUSTADD5 Register (Offset = 114h) [reset = 0h]

TPTC2WRMPUSTADD5 is shown in Figure 4-1001 and described in Table 4-1046.

Return to Summary Table.

Figure 4-417. TPTC2WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD5																															
R/W-0h																															

Table 4-443. TPTC2WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC2

4.7.5.7 TPTC2WRMPUENDADD0 Register (Offset = 120h) [reset = 0h]

TPTC2WRMPUENDADD0 is shown in Figure 4-1002 and described in Table 4-1047.

Return to Summary Table.

Figure 4-418. TPTC2WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD0																															
R/W-0h																															

Table 4-444. TPTC2WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC2

4.7.5.8 TPTC2WRMPUENDADD1 Register (Offset = 124h) [reset = 0h]

TPTC2WRMPUENDADD1 is shown in Figure 4-1003 and described in Table 4-1048.

Return to Summary Table.

Figure 4-419. TPTC2WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD1																															
R/W-0h																															

Table 4-445. TPTC2WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC2

4.7.5.9 TPTC2WRMPUENDADD2 Register (Offset = 128h) [reset = 0h]

TPTC2WRMPUENDADD2 is shown in Figure 4-1004 and described in Table 4-1049.

Return to Summary Table.

Figure 4-420. TPTC2WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD2																															
R/W-0h																															

Table 4-446. TPTC2WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC2

4.7.5.10 TPTC2WRMPUENDADD3 Register (Offset = 12Ch) [reset = 0h]

TPTC2WRMPUENDADD3 is shown in Figure 4-1005 and described in Table 4-1050.

Return to Summary Table.

Figure 4-421. TPTC2WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD3																															
R/W-0h																															

Table 4-447. TPTC2WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC2

4.7.5.11 TPTC2WRMPUENDADD4 Register (Offset = 130h) [reset = 0h]

TPTC2WRMPUENDADD4 is shown in Figure 4-1006 and described in Table 4-1051.

Return to Summary Table.

Figure 4-422. TPTC2WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD4																															
R/W-0h																															

Table 4-448. TPTC2WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC2

4.7.5.12 TPTC2WRMPUENDADD5 Register (Offset = 134h) [reset = 0h]

TPTC2WRMPUENDADD5 is shown in Figure 4-1007 and described in Table 4-1052.

Return to Summary Table.

Figure 4-423. TPTC2WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD5																															
R/W-0h																															

Table 4-449. TPTC2WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC2

4.7.5.13 TPTC2WRMPUERRADD Register (Offset = 140h) [reset = 0h]

TPTC2WRMPUERRADD is shown in Figure 4-1008 and described in Table 4-1053.

Return to Summary Table.

Figure 4-424. TPTC2WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUERRADD																															
R-0h																															

Table 4-450. TPTC2WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC2

4.7.5.14 TPTC2RDMPUSTADD0 Register (Offset = 148h) [reset = 0h]

TPTC2RDMPUSTADD0 is shown in Figure 4-1009 and described in Table 4-1054.

Return to Summary Table.

Figure 4-425. TPTC2RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD0																															
R/W-0h																															

Table 4-451. TPTC2RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC2

4.7.5.15 TPTC2RDMPUSTADD1 Register (Offset = 14Ch) [reset = 0h]

TPTC2RDMPUSTADD1 is shown in Figure 4-1010 and described in Table 4-1055.

Return to Summary Table.

Figure 4-426. TPTC2RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD1																															
R/W-0h																															

Table 4-452. TPTC2RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC2

4.7.5.16 TPTC2RDMPUSTADD2 Register (Offset = 150h) [reset = 0h]

TPTC2RDMPUSTADD2 is shown in Figure 4-1011 and described in Table 4-1056.

Return to Summary Table.

Figure 4-427. TPTC2RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD2																															
R/W-0h																															

Table 4-453. TPTC2RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC2

4.7.5.17 TPTC2RDMPUSTADD3 Register (Offset = 154h) [reset = 0h]

TPTC2RDMPUSTADD3 is shown in Figure 4-1012 and described in Table 4-1057.

Return to Summary Table.

Figure 4-428. TPTC2RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD3																															
R/W-0h																															

Table 4-454. TPTC2RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC2

4.7.5.18 TPTC2RDMPUSTADD4 Register (Offset = 158h) [reset = 0h]

TPTC2RDMPUSTADD4 is shown in Figure 4-1013 and described in Table 4-1058.

Return to Summary Table.

Figure 4-429. TPTC2RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD4																															
R/W-0h																															

Table 4-455. TPTC2RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC2

4.7.5.19 TPTC2RDMPUSTADD5 Register (Offset = 15Ch) [reset = 0h]

TPTC2RDMPUSTADD5 is shown in Figure 4-1014 and described in Table 4-1059.

Return to Summary Table.

Figure 4-430. TPTC2RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD5																															
R/W-0h																															

Table 4-456. TPTC2RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC2

4.7.5.20 TPTC2RDMPUENDADD0 Register (Offset = 168h) [reset = 0h]

TPTC2RDMPUENDADD0 is shown in Figure 4-1015 and described in Table 4-1060.

Return to Summary Table.

Figure 4-431. TPTC2RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD0																															
R/W-0h																															

Table 4-457. TPTC2RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC2

4.7.5.21 TPTC2RDMPUENDADD1 Register (Offset = 16Ch) [reset = 0h]

TPTC2RDMPUENDADD1 is shown in Figure 4-1016 and described in Table 4-1061.

Return to Summary Table.

Figure 4-432. TPTC2RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD1																															
R/W-0h																															

Table 4-458. TPTC2RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC2

4.7.5.22 TPTC2RDMPUENDADD2 Register (Offset = 170h) [reset = 0h]

TPTC2RDMPUENDADD2 is shown in Figure 4-1017 and described in Table 4-1062.

Return to Summary Table.

Figure 4-433. TPTC2RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD2																															
R/W-0h																															

Table 4-459. TPTC2RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC2

4.7.5.23 TPTC2RDMPUENDADD3 Register (Offset = 174h) [reset = 0h]

TPTC2RDMPUENDADD3 is shown in Figure 4-1018 and described in Table 4-1063.

Return to Summary Table.

Figure 4-434. TPTC2RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD3																															
R/W-0h																															

Table 4-460. TPTC2RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC2

4.7.5.24 TPTC2RDMPUENDADD4 Register (Offset = 178h) [reset = 0h]

TPTC2RDMPUENDADD4 is shown in Figure 4-1019 and described in Table 4-1064.

Return to Summary Table.

Figure 4-435. TPTC2RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD4																															
R/W-0h																															

Table 4-461. TPTC2RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC2

4.7.5.25 TPTC2RDMPUENDADD5 Register (Offset = 17Ch) [reset = 0h]

TPTC2RDMPUENDADD5 is shown in Figure 4-1020 and described in Table 4-1065.

Return to Summary Table.

Figure 4-436. TPTC2RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD5																															
R/W-0h																															

Table 4-462. TPTC2RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC2

4.7.5.26 TPTC2RDMPUERRADD Register (Offset = 188h) [reset = 0h]

TPTC2RDMPUERRADD is shown in Figure 4-1021 and described in Table 4-1066.

Return to Summary Table.

Figure 4-437. TPTC2RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUERRADD																															
R-0h																															

Table 4-463. TPTC2RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC2

4.7.5.27 TPTC3WRMPUSTADD0 Register (Offset = 18Ch) [reset = 0h]

TPTC3WRMPUSTADD0 is shown in Figure 4-1022 and described in Table 4-1067.

Return to Summary Table.

Figure 4-438. TPTC3WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD0																															
R/W-0h																															

Table 4-464. TPTC3WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC3

4.7.5.28 TPTC3WRMPUSTADD1 Register (Offset = 190h) [reset = 0h]

TPTC3WRMPUSTADD1 is shown in Figure 4-1023 and described in Table 4-1068.

Return to Summary Table.

Figure 4-439. TPTC3WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD1																															
R/W-0h																															

Table 4-465. TPTC3WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC3

4.7.5.29 TPTC3WRMPUSTADD2 Register (Offset = 194h) [reset = 0h]

TPTC3WRMPUSTADD2 is shown in Figure 4-1024 and described in Table 4-1069.

Return to Summary Table.

Figure 4-440. TPTC3WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD2																															
R/W-0h																															

Table 4-466. TPTC3WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC3

4.7.5.30 TPTC3WRMPUSTADD3 Register (Offset = 198h) [reset = 0h]

TPTC3WRMPUSTADD3 is shown in Figure 4-1025 and described in Table 4-1070.

Return to Summary Table.

Figure 4-441. TPTC3WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD3																															
R/W-0h																															

Table 4-467. TPTC3WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC3

4.7.5.31 TPTC3WRMPUSTADD4 Register (Offset = 19Ch) [reset = 0h]

TPTC3WRMPUSTADD4 is shown in Figure 4-1026 and described in Table 4-1071.

Return to Summary Table.

Figure 4-442. TPTC3WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD4																															
R/W-0h																															

Table 4-468. TPTC3WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC3

4.7.5.32 TPTC3WRMPUSTADD5 Register (Offset = 1A0h) [reset = 0h]

TPTC3WRMPUSTADD5 is shown in Figure 4-1027 and described in Table 4-1072.

Return to Summary Table.

Figure 4-443. TPTC3WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD5																															
R/W-0h																															

Table 4-469. TPTC3WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC3

4.7.5.33 TPTC3WRMPUENDADD0 Register (Offset = 1ACh) [reset = 0h]

TPTC3WRMPUENDADD0 is shown in Figure 4-1028 and described in Table 4-1073.

Return to Summary Table.

Figure 4-444. TPTC3WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD0																															
R/W-0h																															

Table 4-470. TPTC3WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC3

4.7.5.34 TPTC3WRMPUENDADD1 Register (Offset = 1B0h) [reset = 0h]

TPTC3WRMPUENDADD1 is shown in Figure 4-1029 and described in Table 4-1074.

Return to Summary Table.

Figure 4-445. TPTC3WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD1																															
R/W-0h																															

Table 4-471. TPTC3WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC3

4.7.5.35 TPTC3WRMPUENDADD2 Register (Offset = 1B4h) [reset = 0h]

TPTC3WRMPUENDADD2 is shown in Figure 4-1030 and described in Table 4-1075.

Return to Summary Table.

Figure 4-446. TPTC3WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD2																															
R/W-0h																															

Table 4-472. TPTC3WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC3

4.7.5.36 TPTC3WRMPUENDADD3 Register (Offset = 1B8h) [reset = 0h]

TPTC3WRMPUENDADD3 is shown in Figure 4-1031 and described in Table 4-1076.

Return to Summary Table.

Figure 4-447. TPTC3WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD3																															
R/W-0h																															

Table 4-473. TPTC3WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC3

4.7.5.37 TPTC3WRMPUENDADD4 Register (Offset = 1BCh) [reset = 0h]

TPTC3WRMPUENDADD4 is shown in Figure 4-1032 and described in Table 4-1077.

Return to Summary Table.

Figure 4-448. TPTC3WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD4																															
R/W-0h																															

Table 4-474. TPTC3WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC3

4.7.5.38 TPTC3WRMPUENDADD5 Register (Offset = 1C0h) [reset = 0h]

TPTC3WRMPUENDADD5 is shown in Figure 4-1033 and described in Table 4-1078.

Return to Summary Table.

Figure 4-449. TPTC3WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD5																															
R/W-0h																															

Table 4-475. TPTC3WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC3

4.7.5.39 TPTC3WRMPUERRADD Register (Offset = 1CCh) [reset = 0h]

TPTC3WRMPUERRADD is shown in Figure 4-1034 and described in Table 4-1079.

Return to Summary Table.

Figure 4-450. TPTC3WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUERRADD																															
R-0h																															

Table 4-476. TPTC3WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC3

4.7.5.40 TPTC3RDMPUSTADD0 Register (Offset = 1D0h) [reset = 0h]

TPTC3RDMPUSTADD0 is shown in Figure 4-1035 and described in Table 4-1080.

Return to Summary Table.

Figure 4-451. TPTC3RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD0																															
R/W-0h																															

Table 4-477. TPTC3RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC3

4.7.5.41 TPTC3RDMPUSTADD1 Register (Offset = 1D4h) [reset = 0h]

TPTC3RDMPUSTADD1 is shown in Figure 4-1036 and described in Table 4-1081.

Return to Summary Table.

Figure 4-452. TPTC3RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD1																															
R/W-0h																															

Table 4-478. TPTC3RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC3

4.7.5.42 TPTC3RDMPUSTADD2 Register (Offset = 1D8h) [reset = 0h]

TPTC3RDMPUSTADD2 is shown in Figure 4-1037 and described in Table 4-1082.

Return to Summary Table.

Figure 4-453. TPTC3RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD2																															
R/W-0h																															

Table 4-479. TPTC3RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC3

4.7.5.43 TPTC3RDMPUSTADD3 Register (Offset = 1DCh) [reset = 0h]

TPTC3RDMPUSTADD3 is shown in Figure 4-1038 and described in Table 4-1083.

Return to Summary Table.

Figure 4-454. TPTC3RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD3																															
R/W-0h																															

Table 4-480. TPTC3RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC3

4.7.5.44 TPTC3RDMPUSTADD4 Register (Offset = 1E0h) [reset = 0h]

TPTC3RDMPUSTADD4 is shown in Figure 4-1039 and described in Table 4-1084.

Return to Summary Table.

Figure 4-455. TPTC3RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD4																															
R/W-0h																															

Table 4-481. TPTC3RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC3

4.7.5.45 TPTC3RDMPUSTADD5 Register (Offset = 1E4h) [reset = 0h]

TPTC3RDMPUSTADD5 is shown in Figure 4-1040 and described in Table 4-1085.

Return to Summary Table.

Figure 4-456. TPTC3RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD5																															
R/W-0h																															

Table 4-482. TPTC3RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC3

4.7.5.46 TPTC3RDMPUENDADD0 Register (Offset = 1F0h) [reset = 0h]

TPTC3RDMPUENDADD0 is shown in Figure 4-1041 and described in Table 4-1086.

Return to Summary Table.

Figure 4-457. TPTC3RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD0																															
R/W-0h																															

Table 4-483. TPTC3RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC3

4.7.5.47 TPTC3RDMPUENDADD1 Register (Offset = 1F4h) [reset = 0h]

TPTC3RDMPUENDADD1 is shown in Figure 4-1042 and described in Table 4-1087.

Return to Summary Table.

Figure 4-458. TPTC3RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD1																															
R/W-0h																															

Table 4-484. TPTC3RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC3

4.7.5.48 TPTC3RDMPUENDADD2 Register (Offset = 1F8h) [reset = 0h]

TPTC3RDMPUENDADD2 is shown in Figure 4-1043 and described in Table 4-1088.

Return to Summary Table.

Figure 4-459. TPTC3RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD2																															
R/W-0h																															

Table 4-485. TPTC3RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC3

4.7.5.49 TPTC3RDMPUENDADD3 Register (Offset = 1FCh) [reset = 0h]

TPTC3RDMPUENDADD3 is shown in Figure 4-1044 and described in Table 4-1089.

Return to Summary Table.

Figure 4-460. TPTC3RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD3																															
R/W-0h																															

Table 4-486. TPTC3RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC3

4.7.5.50 TPTC3RDMPUENDADD4 Register (Offset = 200h) [reset = 0h]

TPTC3RDMPUENDADD4 is shown in Figure 4-1045 and described in Table 4-1090.

Return to Summary Table.

Figure 4-461. TPTC3RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD4																															
R/W-0h																															

Table 4-487. TPTC3RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC3

4.7.5.51 TPTC3RDMPUENDADD5 Register (Offset = 204h) [reset = 0h]

TPTC3RDMPUENDADD5 is shown in Figure 4-1046 and described in Table 4-1091.

Return to Summary Table.

Figure 4-462. TPTC3RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD5																															
R/W-0h																															

Table 4-488. TPTC3RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC3

4.7.5.52 TPTC3RDMPUERRADD Register (Offset = 210h) [reset = 0h]

TPTC3RDMPUERRADD is shown in Figure 4-1047 and described in Table 4-1092.

Return to Summary Table.

Figure 4-463. TPTC3RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUERRADD																															
R-0h																															

Table 4-489. TPTC3RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC3

4.7.5.53 TPTCMPUVALIDCFG2 Register (Offset = 214h) [reset = 0h]

TPTCMPUVALIDCFG2 is shown in Figure 4-1048 and described in Table 4-1093.

Return to Summary Table.

Figure 4-464. TPTCMPUVALIDCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPTC3RDMPURNGVLD								TPTC3WRMPURNGVLD							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPURNGVLD								TPTC2WRMPURNGVLD							
R/W-0h								R/W-0h							

Table 4-490. TPTCMPUVALIDCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TPTC3RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC3. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
23-16	TPTC3WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC3. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
15-8	TPTC2RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC2. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
7-0	TPTC2WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC2. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled

4.7.5.54 TPTCMPUENCFG2 Register (Offset = 218h) [reset = 0h]

TPTCMPUENCFG2 is shown in Figure 4-1049 and described in Table 4-1094.

Return to Summary Table.

Figure 4-465. TPTCMPUENCFG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPTC3RDMPUERRCLR	TPTC3WRMPUERRCLR	TPTC2RDMPUERRCLR	TPTC2WRMPUERRCLR	TPTC3RDMPUEN	TPTC3WRMPUEN	TPTC2RDMPUEN	TPTC2WRMPUEN
0h	0h	0h	0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-491. TPTCMPUENCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPTC3RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC3. Write 0x1 to clear the MPU error
6	TPTC3WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC3. Write 0x1 to clear the MPU error
5	TPTC2RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC2. Write 0x1 to clear the MPU error
4	TPTC2WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC2. Write 0x1 to clear the MPU error
3	TPTC3RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC3. 0 : MPU is disabled 1 : MPU is enabled
2	TPTC3WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC3. 0 : MPU is disabled 1 : MPU is enabled
1	TPTC2RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC2. 0 : MPU is disabled 1 : MPU is enabled
0	TPTC2WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC2. 0 : MPU is disabled 1 : MPU is enabled

4.7.5.55 L3ECCCFG1 Register (Offset = 268h) [reset = 0h]

L3ECCCFG1 is shown in Figure 4-1050 and described in Table 4-1095.

Return to Summary Table.

Figure 4-466. L3ECCCFG1 Register

31	30	29	28	27	26	25	24
NU					L3ECCREPAIREDBIT		
R-0h					R-0h		
23	22	21	20	19	18	17	16
L3ECCREPAIREDBIT							
R-0h							
15	14	13	12	11	10	9	8
L3ECCREPAIREDBIT							
R-0h							
7	6	5	4	3	2	1	0
L3ECCREPAIREDBIT					L3ECCERRSTAT	L3ECCERRCLR	L3ECCEN
R-0h					R-0h	0h	R/W-0h

Table 4-492. L3ECCCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU	R	0h	Not used
26-3	L3ECCREPAIREDBIT	R	0h	Bit position of repaired bit in L3 ECC memory. Each 6 bits out is this register maps to the corresponding 32 bit location in the data.
2	L3ECCERRSTAT	R	0h	Latched status for L3 ECC error.
1	L3ECCERRCLR		0h	Clear bit for L3 ECC.
0	L3ECCEN	R/W	0h	Enable for L3 ECC logic

4.7.5.56 L3ECCCFG2 Register (Offset = 26Ch) [reset = 0h]

L3ECCCFG2 is shown in Figure 4-1051 and described in Table 4-1096.

Return to Summary Table.

Figure 4-467. L3ECCCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU															L3ECCFAULTADDR																
R-0h															R-0h																

Table 4-493. L3ECCCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU	R	0h	Not used
16-0	L3ECCFAULTADDR	R	0h	Fault address of L3 ECC memory.

4.7.5.57 DSS2MSSSWIRQ Register (Offset = 270h) [reset = 0h]

DSS2MSSSWIRQ is shown in Figure 4-1052 and described in Table 4-1097.

Return to Summary Table.

Figure 4-468. DSS2MSSSWIRQ Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU						MSSSWIRQ2	MSSSWIRQ1
R-0h						0h	0h

Table 4-494. DSS2MSSSWIRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	0h	Not used
1	MSSSWIRQ2		0h	single bit and self clearing interrupt. writing a '1' to this bit will generate a pulse from DSS to MSS VIM line 61
0	MSSSWIRQ1		0h	single bit and self clearing interrupt. writing a '1' to this bit will generate a pulse from DSS to MSS VIM line 52

4.8 68xx/64xx Control Registers

4.8.1 MSS_TOPRCM Registers

Table 4-802 lists the memory-mapped registers for the MSS_TOPRCM. All register offset addresses not listed in Table 4-802 should be considered as reserved locations and the register contents should not be modified.

Table 4-495. MSS_TOPRCM Registers

Offset	Acronym	Register Name	Section
8h	BSSCTL	Control Signals to BSS	Section 4.9.1.1
Ch	DSSCTL	Control Signals to DSS	Section 4.9.1.2
10h	EXTCLKDIV	Clock divide value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.3
14h	EXTCLKSRCSEL	Clock source select value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.4
18h	EXTCLKCTL	Clock gate control for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.5
1Ch	SOFTSYSRST	Software triggered Warm Reset	Section 4.9.1.6
20h	WDRSTEN	Issue Warm reset upon MSS Watch dog reset	Section 4.9.1.7
24h	SYSRSTCAUSE	Reset cause register	Section 4.9.1.8
28h	SYSRSTCAUSECLR	Clear Reset Cause register	Section 4.9.1.9
34h	MISCCAPT	Capture required Status values across the chip.	Section 4.9.1.10
38h	DCDCCTL0	PMIC_CLKOUT dethering control	Section 4.9.1.11
3Ch	DCDCCTL1	PMIC_CLKOUT dethering control	Section 4.9.1.12
48h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.1.13
4Ch	LVDSPADCTL0	LVDS pad control	Section 4.9.1.14
50h	LVDSPADCTL1	LVDS pad control	Section 4.9.1.15
60h	DFTREG0		Section 4.9.1.16
64h	DFTREG1		Section 4.9.1.17
74h	DFTREG5		Section 4.9.1.18
DCh	MEMINITDONE	Memory Initialization done status for memories in BSS and DSS	Section 4.9.1.19
ECh	SPARE0_3	Spare registers 0 to 3	Section 4.9.1.20
FCh	MSS_SIGNATURE	MSS Signature register	Section 4.9.1.21
158h	GEMBOOTSTCEN		Section 4.9.1.22
178h	MISCCTL1	Miscellaneous Control Register	Section 4.9.1.23
180h	USERMODEEN2	USER_MODE_ACCESS_EN	Section 4.9.1.24
18Ch	SYSTICK		Section 4.9.1.25
1C4h	SECURECFGREG1		Section 4.9.1.26
1C8h	SECURECFGREG2		Section 4.9.1.27
1CCh	SECURECFGREG3		Section 4.9.1.28
1D0h	SECURECFGREG4		Section 4.9.1.29
1D4h	SECURERAMREG		Section 4.9.1.30
1E4h	SPAREMULTIBIT	MIBSPI setup register	Section 4.9.1.31
1E8h	SPARE4_9	Spare registers 4 to 9	Section 4.9.1.32
200h	UID31TO0	Efuse Row Capture register for FROM1	Section 4.9.1.33
204h	UID63TO32	Efuse Row Capture register for FROM1	Section 4.9.1.34
208h	UID95TO64	Efuse Row Capture register for FROM1	Section 4.9.1.35
20Ch	UID119TO96	Efuse Row Capture register for FROM1	Section 4.9.1.36
2A8h	MEMINITSTARTSHMEM	Shared memory initialization start	Section 4.9.1.37
2ACh	MEMINITDONESHMEM	Shared memory intialization end	Section 4.9.1.38
2B0h	DSSMEMTAB0	Controls ordering of banks in shared memory associated with DSS	Section 4.9.1.39
2BCh	TCMAMEMTAB	Controls ordering of banks in shared memory associated with MSS TCMA	Section 4.9.1.40
2C0h	TCMBMEMTAB	Controls ordering of banks in shared memory associated with MSS TCMB	Section 4.9.1.41
2C8h	SHMEMBANKSEL3TO0	Shared memory master allocation.	Section 4.9.1.42

Table 4-495. MSS_TOPRCM Registers (continued)

Offset	Acronym	Register Name	Section
2CCh	SHMEMBANKSEL7TO4	Shared memory master allocation.	Section 4.9.1.43
2D0h	PBISTCLKCTL	PBIST clock control register	Section 4.9.1.44

Complex bit access types are encoded to fit into small table cells. Table 4-803 shows the codes that are used for access types in this section.

Table 4-496. MSS_TOPRCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.8.1.1 BSSCTL Register (Offset = 8h) [reset = ADADADADh]

BSSCTL is shown in Figure 4-767 and described in Table 4-804.

Return to Summary Table.

Control Signals to BSS

Figure 4-469. BSSCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BSSCPUHALT								RESERVED							
R/W-ADh								R/W-ADh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
R/W-ADh															

Table 4-497. BSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BSSCPUHALT	R/W	ADh	Halt BSS CR4, To halt : either 3:0 should be 0xD or 7:4 should be 0xA One should Halt the processor before releasing BSS reset.
23-0	RESERVED	R/W	ADh	Reserved

4.8.1.2 DSSCTL Register (Offset = Ch) [reset = 00ADADADh]

DSSCTL is shown in Figure 4-768 and described in Table 4-805.

Return to Summary Table.

Control Signals to DSS

Figure 4-470. DSSCTL Register

31	30	29	28	27	26	25	24
NU					GEMLRSTN	GEMGRSTN	GEMPORZ
0h					R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-ADh							
15	14	13	12	11	10	9	8
RESERVED							
R/W-ADh							
7	6	5	4	3	2	1	0
RESERVED							
R/W-ADh							

Table 4-498. DSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU		0h	Reserved
26	GEMLRSTN	R/W	0h	DSP Local Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
25	GEMGRSTN	R/W	0h	DSP Global Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
24	GEMPORZ	R/W	0h	DSP Por Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
23-0	RESERVED	R/W	ADh	Reserved

4.8.1.3 EXTCLKDIV Register (Offset = 10h) [reset = 0h]

EXTCLKDIV is shown in Figure 4-769 and described in Table 4-806.

Return to Summary Table.

Clock divide value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-471. EXTCLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2DIV								EXTCLK1DIV							
R/W-0h								R/W-0h							

Table 4-499. EXTCLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Reserved
15-8	EXTCLK2DIV	R/W	0h	Divide value for clock source (selected by field EXTCLKSRCSEL in register CLKSRCSEL.) to PMIC_CLKOUT generation logic “0000_0000” => div1 “0000_0001” => div2 | | “1111_1111” => div256 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming EXTCLK2SRCSEL.
7-0	EXTCLK1DIV	R/W	0h	Divide value for MCU_CLKOUT (the one going out of chip) source clock selected by field EXTCLKSRCSEL in register CLKSRCSEL. “0000_0000” => div1 “0000_0001” => div2 | | “1111_1111” => div256 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming EXTCLK1SRCSEL.

4.8.1.4 EXTCLKSRCSEL Register (Offset = 14h) [reset = 0h]

EXTCLKSRCSEL is shown in Figure 4-770 and described in Table 4-807.

Return to Summary Table.

Clock source select value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-472. EXTCLKSRCSEL Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1				EXTCLK2SRCSEL			
0h				R/W-0h			
7	6	5	4	3	2	1	0
NU0				EXTCLK1SRCSEL			
0h				R/W-0h			

Table 4-500. EXTCLKSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU1		0h	Reserved
11-8	EXTCLK2SRCSEL	R/W	0h	Select clock source for PMIC_CLKOUT generation “000” => CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) “001” => RCCLK (10Mhz) “010” => 600Mhz PLL divided clock “011” => 240Mhz PLL divided clock “100” => RCCLK (10Mhz) “101” => RCCLK (10Mhz) “110” => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mhz) “111” => RCCLK
7-4	NU0		0h	Reserved
3-0	EXTCLK1SRCSEL	R/W	0h	Select clock source for MCU_CLKOUT “000” =>CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) “001” => RCCLK (10Mhz) “010” => 600Mhz PLL divided clock “011” => 240Mhz PLL divided clock “100” => RCCLK (10Mhz) “101” => RCCLK (10Mhz) “110” => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mhz) “111” => RCCLK

4.8.1.5 EXTCLKCTL Register (Offset = 18h) [reset = ADADh]

EXTCLKCTL is shown in Figure 4-771 and described in Table 4-808.

Return to Summary Table.

Clock gate control for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-473. EXTCLKCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2GATE								EXTCLK1GATE							
R/W-ADh								R/W-ADh							

Table 4-501. EXTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Reserved
15-8	EXTCLK2GATE	R/W	ADh	Pre clock divider gate for PMIC_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA
7-0	EXTCLK1GATE	R/W	ADh	Pre clock divider gate for MCU_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA

4.8.1.6 SOFTSYSRST Register (Offset = 1Ch) [reset = 0h]

SOFTSYSRST is shown in Figure 4-772 and described in Table 4-809.

Return to Summary Table.

Software triggered Warm Reset

Figure 4-474. SOFTSYSRST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SOFTSYSRST							
0h								R/W-0h							

Table 4-502. SOFTSYSRST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	SOFTSYSRST	R/W	0h	Write 0xAD to trigger warm reset to the chip.

4.8.1.7 WDRSTEN Register (Offset = 20h) [reset = 0h]

WDRSTEN is shown in Figure 4-773 and described in Table 4-810.

Return to Summary Table.

Issue Warm reset upon MSS Watch dog reset

Figure 4-475. WDRSTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								WDRSTEN							
0h																								R/W-0h							

Table 4-503. WDRSTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	WDRSTEN	R/W	0h	Write 0xAD to trigger warm reset to the chip upon MSS wdog reset.

4.8.1.8 SYSRSTCAUSE Register (Offset = 24h) [reset = 0h]

SYSRSTCAUSE is shown in Figure 4-774 and described in Table 4-811.

Return to Summary Table.

Reset cause register

Figure 4-476. SYSRSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU												SYSRSTCAUSE			
0h												R-0h			

Table 4-504. SYSRSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU		0h	Reserved
3-0	SYSRSTCAUSE	R	0h	Gives cause of chip reset “1001” : System out of NRESET “1010” : Warm reset because of MSS Wdog. “1100” : Warm reset because of Software trigger- SOFTSYSRST “1000” : External Warm Reset Note: The ROM Bootloader clears this register as part of its processing, so it always reads zero by the application or secondary bootloader. However, the ROM Bootloader saves the original value into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.8.1.9 SYSRSTCAUSECLR Register (Offset = 28h) [reset = 0h]

SYSRSTCAUSECLR is shown in Figure 4-775 and described in Table 4-812.

Return to Summary Table.

Clear Reset Cause register

Figure 4-477. SYSRSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SYSRSTCAUSECLR							
0h								0h							

Table 4-505. SYSRSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	SYSRSTCAUSECLR		0h	Write 0xAD to clear SYSRSTCAUSE..Self clearing

4.8.1.10 MISCCAPT Register (Offset = 34h) [reset = 0h]

MISCCAPT is shown in Figure 4-776 and described in Table 4-813.

Return to Summary Table.

Capture required Status values across the chip.

Figure 4-478. MISCCAPT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCAPT																															
R-0h																															

Table 4-506. MISCCAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCAPT	R	0h	0: No error Any other non-zero value: Error

4.8.1.11 DCDCCTL0 Register (Offset = 38h) [reset = 0h]

DCDCCTL0 is shown in Figure 4-777 and described in Table 4-814.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-479. DCDCCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCDCCTL0																															
R/W-0h																															

Table 4-507. DCDCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DCDCCTL0	R/W	0h	[26:0] PMIC_CLKOUT Frequency slope value.Unsigned

4.8.1.12 DCDCCTL1 Register (Offset = 3Ch) [reset = 0h]

DCDCCTL1 is shown in Figure 4-778 and described in Table 4-815.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-480. DCDCCTL1 Register

31	30	29	28	27	26	25	24
DCDCCTL1							
R/W-0h							
23	22	21	20	19	18	17	16
DCDCCTL1							
R/W-0h							
15	14	13	12	11	10	9	8
DCDCCTL1							
R/W-0h							
7	6	5	4	3	2	1	0
DCDCCTL1						DCDCLKEN	DCDCRST
R/W-0h						R/W-0h	R/W-0h

Table 4-508. DCDCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	DCDCCTL1	R/W	0h	PMIC_CLKOUT dithering control [8] : Frequency accumulation Mode . ‘0’ : Continous mode.’1’ : Staircase mode [9] : ‘1’ Enables dither [23:16] : Minimum frequency threshold [31:24] : Maximum frequency threshold
1	DCDCLKEN	R/W	0h	PMIC_CLKOUT Enable – Multi Bit
0	DCDCRST	R/W	0h	PMIC_CLKOUT dithering control block reset (active high) – Multi Bit

4.8.1.13 USERMODEEN Register (Offset = 48h) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-481. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-509. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space from 0x00 to 0XFF

4.8.1.14 LVDSPADCTL0 Register (Offset = 4Ch) [reset = 01010101h]

LVDSPADCTL0 is shown in Figure 4-780 and described in Table 4-817.

Return to Summary Table.

LVDS pad control

Figure 4-482. LVDSPADCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL0																															
R/W-01010101h																															

Table 4-510. LVDSPADCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL0	R/W	01010101h	0 : pwrdn Control for i_LVDSclk_io_cell 1 : lopwra Control for i_LVDSclk_io_cell 2 : lopwrb Control for i_LVDSclk_io_cell 3 : lpsel Control for i_LVDSclk_io_cell 4 : sub_lvds_en Control for i_LVDSclk_io_cell 5 : hiz_disable Control for i_LVDSclk_io_cell 6 : ext_res_en Control for i_LVDSclk_io_cell 7 : Reserved 8 : pwrdn Control for i_LVDS_tx0_io_cell 9 : lopwra Control for i_LVDS_tx0_io_cell 10 : lopwrb Control for i_LVDS_tx0_io_cell 11 : lpsel Control for i_LVDS_tx0_io_cell 12 : sub_lvds_en Control for i_LVDS_tx0_io_cell 13 : hiz_disable Control for i_LVDS_tx0_io_cell 14 : ext_res_en Control for i_LVDS_tx0_io_cell 15 : Reserved 16 : pwrdn Control for i_LVDS_tx1_io_cell 17 : lopwra Control for i_LVDS_tx1_io_cell 18 : lopwrb Control for i_LVDS_tx1_io_cell 19 : lpsel Control for i_LVDS_tx1_io_cell 20 : sub_lvds_en Control for i_LVDS_tx1_io_cell 21 : hiz_disable Control for i_LVDS_tx1_io_cell 22 : ext_res_en Control for i_LVDS_tx1_io_cell 23 -31 : Reserved

4.8.1.15 LVDSPADCTL1 Register (Offset = 50h) [reset = 101h]

LVDSPADCTL1 is shown in Figure 4-781 and described in Table 4-818.

Return to Summary Table.

LVDS pad control

Figure 4-483. LVDSPADCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL1																															
R/W-101h																															

Table 4-511. LVDSPADCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL1	R/W	101h	0- 7 : Reserved 8 : pwrdn Control for i_LVDSfrclk_io_cell 9 : lopwra Control for i_LVDSfrclk_io_cell 10 : lopwrb Control for i_LVDSfrclk_io_cell 11 : lpsel Control for i_LVDSfrclk_io_cell 12 : sub_lvds_en Control for i_LVDSfrclk_io_cell 13 : hiz_disable Control for i_LVDSfrclk_io_cell 14 : ext_res_en Control for i_LVDSfrclk_io_cell 15-23: Reserved 24 : pwrdn Control for lvds_bias_cell 25 : efuse_set Control for lvds_bias_cell

4.8.1.16 DFTREG0 Register (Offset = 60h) [reset = 0h]

DFTREG0 is shown in Figure 4-782 and described in Table 4-819.

Return to Summary Table.

Figure 4-484. DFTREG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG0																															
R/W-0h																															

Table 4-512. DFTREG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG0	R/W	0h	[3:0] : MSS PBIST SELFTEST KEY = 4'b1010 [4] : Reserved [5] : PBIST IP reset control. 1 = reset, 0 = out of reset. The PBIST_SELFTEST_KEY must be enabled to program this bit. [31:6] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.8.1.17 DFTREG1 Register (Offset = 64h) [reset = 0h]

DFTREG1 is shown in Figure 4-783 and described in Table 4-820.

Return to Summary Table.

Figure 4-485. DFTREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG1																															
R/W-0h																															

Table 4-513. DFTREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG1	R/W	0h	[31:0] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.8.1.18 DFTREG5 Register (Offset = 74h) [reset = 0h]

DFTREG5 is shown in Figure 4-784 and described in Table 4-821.

Return to Summary Table.

Figure 4-486. DFTREG5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG5																															
R/W-0h																															

Table 4-514. DFTREG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG5	R/W	0h	[0] : Reserved [4:1] DSP PBIST SELFTEST KEY = 4'b1010 [8:5]: Subsystem level memory self-test power clock gate enable controls [31:9] : Reserved

4.8.1.19 MEMINITDONE Register (Offset = DCh) [reset = 0h]

MEMINITDONE is shown in Figure 4-822 and described in Table 4-861.

Return to Summary Table.

Memory Initialization done status for memories in BSS and DSS

Figure 4-487. MEMINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED						BSSVIMMEM	RESERVED
0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED							
0h							

Table 4-515. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED		0h	Reserved
9	BSSVIMMEM	R	0h	Memory Initialization done status for BSS VIM memory
8-0	RESERVED	R	0h	Reserved

4.8.1.20 SPARE0_3 Register (Offset = ECh, F0h, F4h, F8h) [reset = 0h]

SPARE0_3 is shown in Figure 4-786 and described in Table 4-823.

Return to Summary Table.

Spare Register

Figure 4-488. SPARE0_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE0_3																															
R/W-0h																															

Table 4-516. SPARE0_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE0_3	R/W	0h	Four 32-bit registers that may be used by an application for any purpose.

4.8.1.21 MSS_SIGNATURE Register (Offset = FCh) [reset = 0BB5202Fh]

MSS_SIGNATURE is shown in Figure 4-787 and described in Table 4-824.

Return to Summary Table.

Spare Register

Figure 4-489. MSS_SIGNATURE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSS_SIGNATURE																															
R/W-0BB5202Fh																															

Table 4-517. MSS_SIGNATURE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSS_SIGNATURE	R/W	0BB5202Fh	

4.8.1.22 GEMBOOTSTCEN Register (Offset = 158h) [reset = 0h]

GEMBOOTSTCEN is shown in Figure 4-788 and described in Table 4-825.

Return to Summary Table.

Figure 4-490. GEMBOOTSTCEN Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							GEMBOOTSTCEN
R-0h							R/W-0h

Table 4-518. GEMBOOTSTCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	
0	GEMBOOTSTCEN	R/W	0h	'1' : Enable GEM STC during GEM power UP

4.8.1.23 MISCCTL1 Register (Offset = 178h) [reset = 0h]

MISCCTL1 is shown in Figure 4-789 and described in Table 4-826.

Return to Summary Table.

Miscellaneous Control Register

Figure 4-491. MISCCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCTL1																															
R/W-0h																															

Table 4-519. MISCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCTL1	R/W	0h	7:0 : Write 0xAD to enable Warm_resetn from external device in addition to internally generated warm reset 16:8 : Write 0xAD to take board level loop back clock for QSPI. 24:16 : Write 0xAD to external clock as QSPI baud clock source – needed for DFT IO char..

4.8.1.24 USERMODEEN2 Register (Offset = 180h) [reset = 0h]

USERMODEEN2 is shown in Figure 4-790 and described in Table 4-827.

Return to Summary Table.

Figure 4-492. USERMODEEN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN2																															
R/W-0h																															

Table 4-520. USERMODEEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN2	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space which are resettable only by Power on reset. i.e. from offset address 0x100 to 0x1FF

4.8.1.25 SYSTICK Register (Offset = 18Ch) [reset = 0h]

SYSTICK is shown in Figure 4-791 and described in Table 4-828.

Return to Summary Table.

Figure 4-493. SYSTICK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSTICK																															
R-0h																															

Table 4-521. SYSTICK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYSTICK	R	0h	Continuous counter running on 32Khz derived from RC clock.

4.8.1.26 SECURECFGREG1 Register (Offset = 1C4h) [reset = 00700777h]

SECURECFGREG1 is shown in Figure 4-792 and described in Table 4-829.

Return to Summary Table.

Figure 4-494. SECURECFGREG1 Register

31	30	29	28	27	26	25	24
RESERVED	JTAGFIREWALLEN			RESERVED			
0h	R/W-0h			R/W-0h			
23	22	21	20	19	18	17	16
RESERVED	SECURERAMFIREWALLEN			RESERVED	LOGGERFIREWALLEN		
0h	R/W-7h			0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	TRACEFIREWALLEN			RESERVED	CRYPTOFIREWALLEN		
0h	R/W-0h			0h	R/W-7h		
7	6	5	4	3	2	1	0
RESERVED	CUSTCEKFIREWALLEN			RESERVED			
0h	R/W-7h			R/W-0h			

Table 4-522. SECURECFGREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED		0h	
30-28	JTAGFIREWALLEN	R/W	0h	JTAG Firewall.Firewall Disabled for value "111" and enabled for rest
27-23	RESERVED		0h	
22-20	SECURERAMFIREWALLEN	R/W	7h	Set Secure RAM Firewall.Firewall Disabled for value "111" and enabled for rest
19	RESERVED		0h	
18-16	LOGGERFIREWALLEN	R/W	0h	Set Logger Firewall.Firewall Disabled for value "111" and enabled for rest
15	RESERVED		0h	
14-12	TRACEFIREWALLEN	R/W	0h	Set Trace Firewall. Firewall Disabled for value "111" and enabled for rest
11	RESERVED		0h	
10-8	CRYPTOFIREWALLEN	R/W	7h	Set Crypto Firewall. Firewall Disabled for value "111" and enabled for rest
7	RESERVED		0h	
6-4	CUSTCEKFIREWALLEN	R/W	7h	Set CEK1,CEK2 firewall.Firewall Disabled for value "111" and enabled for rest
3-0	RESERVED		0h	

4.8.1.27 SECURECFGREG2 Register (Offset = 1C8h) [reset = 0h]

SECURECFGREG2 is shown in Figure 4-793 and described in Table 4-830.

Return to Summary Table.

Figure 4-495. SECURECFGREG2 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED					CUSTKEYERASE		
0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					DMMFIREWALLEN		
0h					R/W-0h		

Table 4-523. SECURECFGREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED		0h	Reserved
10-8	CUSTKEYERASE	R/W	0h	Erase CEK1 ,CEK2,CPK Keys when value "111" is written
7-3	RESERVED		0h	Reserved
2-0	DMMFIREWALLEN	R/W	0h	DMM Firewall.Firewall Disabled for value "111" and enabled for rest

4.8.1.28 SECURECFGREG3 Register (Offset = 1CCh) [reset = 0h]

SECURECFGREG3 is shown in Figure 4-794 and described in Table 4-831.

Return to Summary Table.

Figure 4-496. SECURECFGREG3 Register

31	30	29	28	27	26	25	24
RESERVED	JTAGSTICKYBIT			RESERVED			
0h	R/W-0h			0h			
23	22	21	20	19	18	17	16
RESERVED	SECURERAMSTICKYBIT			RESERVED	TRACESTICKYBIT		
0h	R/W-0h			0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	CRYPTOSTICKYBIT			RESERVED	CUSTCEKSTICKYBIT		
0h	R/W-0h			0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					LOGGERSTICKYBIT		
0h					R/W-0h		

Table 4-524. SECURECFGREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED		0h	
30-28	JTAGSTICKYBIT	R/W	0h	JTAG Sticky Reg. Sticky reg is set when value "111 is written
27-23	RESERVED		0h	
22-20	SECURERAMSTICKYBIT	R/W	0h	Secure RAM Sticky Reg. Sticky reg is set when value "111 is written
19	RESERVED		0h	
18-16	TRACESTICKYBIT	R/W	0h	Trace Sticky Reg. Sticky reg is set when value "111 is written
15	RESERVED		0h	
14-12	CRYPTOSTICKYBIT	R/W	0h	Crypto Sticky Reg. Sticky reg is set when value "111 is written
11	RESERVED		0h	
10-8	CUSTCEKSTICKYBIT	R/W	0h	CEK1,CEK2 Sticky Reg for firewall. Sticky reg is set when value "111 is written
7-3	RESERVED		0h	
2-0	LOGGERSTICKYBIT	R/W	0h	Logger Sticky Reg. Sticky reg is set when value "111 is written

4.8.1.29 SECURECFGREG4 Register (Offset = 1D0h) [reset = 0h]

SECURECFGREG4 is shown in Figure 4-795 and described in Table 4-832.

Return to Summary Table.

Figure 4-497. SECURECFGREG4 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RESERVED					DMMSTICKYBIT		
0h					R/W-0h		

Table 4-525. SECURECFGREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED		0h	Reserved
2-0	DMMSTICKYBIT	R/W	0h	DMM Sticky Reg. Sticky reg is set when value "111 is written

4.8.1.30 SECURERAMREG Register (Offset = 1D4h) [reset = 0h]

SECURERAMREG is shown in Figure 4-796 and described in Table 4-833.

Return to Summary Table.

Figure 4-498. SECURERAMREG Register

31	30	29	28	27	26	25	24
NU4							SECURERAMKEY255
0h							R/W-0h
23	22	21	20	19	18	17	16
SECURERAMKEYIDX							
R/W-0h							
15	14	13	12	11	10	9	8
NU2							SECURERAMKEYRD
0h							0h
7	6	5	4	3	2	1	0
NU10							SECURERAMRDDONE
0h							R-0h

Table 4-526. SECURERAMREG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4		0h	
24	SECURERAMKEY255	R/W	0h	1: Secure RAM key bitwidth = 255 0:Secure RAM key bitwidth = 128
23-16	SECURERAMKEYIDX	R/W	0h	Index to Secure RAM
15-9	NU2		0h	
8	SECURERAMKEYRD		0h	1: Load key from secure RAM into register to be used by AES engine.Self clearing
7-1	NU10		0h	
0	SECURERAMRDDONE	R	0h	Secure RAM key loaded into register

4.8.1.31 SPAREMULTIBIT Register (Offset = 1E4h) [reset = FFFF0000h]

SPAREMULTIBIT is shown in Figure 4-797 and described in Table 4-834.

Return to Summary Table.

Spare Register

Figure 4-499. SPAREMULTIBIT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-1h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-1h							
15	14	13	12	11	10	9	8
RESERVED				SPAREMULTIBIT11	SPAREMULTIBIT10	SPAREMULTIBIT9	SPAREMULTIBIT8
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SPAREMULTIBIT7	SPAREMULTIBIT6	SPAREMULTIBIT5	SPAREMULTIBIT4	SPAREMULTIBIT3	SPAREMULTIBIT2	SPAREMULTIBIT1	SPAREMULTIBIT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-527. SPAREMULTIBIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	1h	Reserved
11	SPAREMULTIBIT11	R/W	0h	MIBSPIB : when set the TRIGGER's are un-gated only when chip-select is active
10	SPAREMULTIBIT10	R/W	0h	SPIB trigger source polarity select.'0' - Polarity 0,'1'-Polarity 1
9	SPAREMULTIBIT9	R/W	0h	SPIA trigger source polarity select.'0' - Polarity 0,'1'-Polarity 1
8	SPAREMULTIBIT8	R/W	0h	'1': MIBSPIB External chip select is overridden with the value of MIBSPIB CS polarity-slave mode
7	SPAREMULTIBIT7	R/W	0h	'1': MIBSPIA External chip select is overridden with the value of MIBSPIA CS polarity-slave mode
6	SPAREMULTIBIT6	R/W	0h	MIBSPIB CS Trigger SRC enable '1': Use CS as trigger source
5	SPAREMULTIBIT5	R/W	0h	MIBSPIB CS polarity-slave mode 1: Active high 0:Active low
4	SPAREMULTIBIT4	R/W	0h	MIBSPIB MISO OE_N Control based on Chip select(CS)-applicable in slave mode 1:MISO OEN controlled based on CS.When CS is inactive OE_N=1 0:MISO OEN controlled by IP
3	SPAREMULTIBIT3	R/W	0h	MIBSPIA :When set the TRIGGER's are un-gated only when chip-select is active.
2	SPAREMULTIBIT2	R/W	0h	MIBSPIA CS Trigger SRC enable-slave mode '1': Use CS as trigger source
1	SPAREMULTIBIT1	R/W	0h	MIBSPIA CS polarity-slave mode 1: Active high 0:Active low
0	SPAREMULTIBIT0	R/W	0h	MIBSPIA MISO OE_N Control based on Chip select(CS)-applicable in slave mode 1:MISO OEN controlled based on CS.When CS is inactive OE_N=1,else controlled by IP 0:MISO OEN controlled by IP

4.8.1.32 SPARE4_9 Register (Offset = 1E8h, 1ECh, … 1FCh) [reset = 0h]

SPARE4_9 is shown in Figure 4-798 and described in Table 4-835.

Return to Summary Table.

Figure 4-500. SPARE4_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID31TO0																															
R-0h																															

Table 4-528. SPARE4_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE4_9	R	0h	Six 32-bit registers that may be used by an application for any purpose. SPARE4 through SPARE8 are not used by the ROM Bootloader, and will survive a Warm Reset. SPARE9 is used by the ROM Bootloader to record RCM_RSTCAUSE and TOPRCM_SYSRSTCAUSE before they are modified by the Bootloader. SPARE 9 is written by the Bootloader thusly: SPARE9[3:0] - current TOPRCM reset cause SPARE9[11:4] - current MSS RCM reset cause SPARE9[15:12] - UNUSED SPARE9[19:16] - previous TOP RCM reset cause SPARE9[27:20] - previous MSS RCM reset cause SPARE9[31:38] - UNUSED

4.8.1.33 UID31TO0 Register (Offset = 200h) [reset = 0h]

UID31TO0 is shown in Figure 4-799 and described in Table 4-836.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-501. UID31TO0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID31TO0																															
R-0h																															

Table 4-529. UID31TO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID31TO0	R	0h	FROM1 Efuse Read UID[31:0]

4.8.1.34 UID63TO32 Register (Offset = 204h) [reset = 0h]

UID63TO32 is shown in Figure 4-800 and described in Table 4-837.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-502. UID63TO32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID63TO32																															
R-0h																															

Table 4-530. UID63TO32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID63TO32	R	0h	FROM1 Efuse Read UID[63:32]

4.8.1.35 UID95TO64 Register (Offset = 208h) [reset = 0h]

UID95TO64 is shown in Figure 4-801 and described in Table 4-838.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-503. UID95TO64 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID95TO64																															
R-0h																															

Table 4-531. UID95TO64 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID95TO64	R	0h	FROM1 Efuse Read UID[95:64]

4.8.1.36 UID119TO96 Register (Offset = 20Ch) [reset = 0h]

UID119TO96 is shown in Figure 4-802 and described in Table 4-839.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-504. UID119TO96 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID119TO96																															
R-0h																															

Table 4-532. UID119TO96 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID119TO96	R	0h	FROM1 Efuse Read UID[119:96]

4.8.1.37 MEMINITSTARTSHMEM Register (Offset = 2A8h) [reset = 0h]

MEMINITSTARTSHMEM is shown in Figure 4-803 and described in Table 4-840.

Return to Summary Table.

Shared memory initialization start

Figure 4-505. MEMINITSTARTSHMEM Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
MEMINITSTARTBANK7	MEMINITSTARTBANK6	MEMINITSTARTBANK5	MEMINITSTARTBANK4	MEMINITSTARTBANK3	MEMINITSTARTBANK2	MEMINITSTARTBANK1	MEMINITSTARTBANK0
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-533. MEMINITSTARTSHMEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1		0h	Not used
7	MEMINITSTARTBANK7		0h	Writing '1' will trigger Shared memory initialization for bank 7.Self clearing
6	MEMINITSTARTBANK6		0h	Writing '1' will trigger Shared memory initialization for bank 6.Self clearing
5	MEMINITSTARTBANK5		0h	Writing '1' will trigger Shared memory initialization for bank 5.Self clearing
4	MEMINITSTARTBANK4		0h	Writing '1' will trigger Shared memory initialization for bank 4.Self clearing
3	MEMINITSTARTBANK3		0h	Writing '1' will trigger Shared memory initialization for bank 3.Self clearing
2	MEMINITSTARTBANK2		0h	Writing '1' will trigger Shared memory initialization for bank 2.Self clearing
1	MEMINITSTARTBANK1		0h	Writing '1' will trigger Shared memory initialization for bank 1.Self clearing
0	MEMINITSTARTBANK0		0h	Writing '1' will trigger Shared memory initialization for bank 0.Self clearing

4.8.1.38 MEMINITDONESHMEM Register (Offset = 2ACh) [reset = 0h]

MEMINITDONESHMEM is shown in Figure 4-804 and described in Table 4-841.

Return to Summary Table.

Shared memory intialization end

Figure 4-506. MEMINITDONESHMEM Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
MEMINITDONEBANK7	MEMINITDONEBANK6	MEMINITDONEBANK5	MEMINITDONEBANK4	MEMINITDONEBANK3	MEMINITDONEBANK2	MEMINITDONEBANK1	MEMINITDONEBANK0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 4-534. MEMINITDONESHMEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1		0h	Not used
7	MEMINITDONEBANK7	R	0h	Memory Initialization done status for Shared memory for bank 7
6	MEMINITDONEBANK6	R	0h	Memory Initialization done status for Shared memory for bank 6
5	MEMINITDONEBANK5	R	0h	Memory Initialization done status for Shared memory for bank 5
4	MEMINITDONEBANK4	R	0h	Memory Initialization done status for Shared memory for bank 4
3	MEMINITDONEBANK3	R	0h	Memory Initialization done status for Shared memory for bank 3
2	MEMINITDONEBANK2	R	0h	Memory Initialization done status for Shared memory for bank 2
1	MEMINITDONEBANK1	R	0h	Memory Initialization done status for Shared memory for bank 1
0	MEMINITDONEBANK0	R	0h	Memory Initialization done status for Shared memory for bank 0

4.8.1.39 DSSMEMTAB0 Register (Offset = 2B0h) [reset = 76543210h]

DSSMEMTAB0 is shown in Figure 4-805 and described in Table 4-842.

Return to Summary Table.

Controls ordering of banks in shared memory associated with DSS

Figure 4-507. DSSMEMTAB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSSMEMTAB0																															
R/W-76543210h																															

Table 4-535. DSSMEMTAB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DSSMEMTAB0	R/W	76543210h	DSS L3RAM memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 0), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 1), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 2), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 3). 4th 128KB address goes to bank number programmed in [19:16] of this register(default is bank 4), 5th 128KB address goes to bank number programmed in [23:20] of this register(default is bank 5), 6th 128KB address goes to bank number programmed in [27:24] of this register(default is bank 6), 7th 128KB address goes to bank number programmed in [31:28] of this register(default is bank 7). Corresponding banks need to be selected for DSS in SHMEMBANKSEL regsiter

4.8.1.40 TCMAMEMTAB Register (Offset = 2BCh) [reset = 7654h]

TCMAMEMTAB is shown in Figure 4-806 and described in Table 4-843.

Return to Summary Table.

Controls ordering of banks in shared memory associated with MSS TCMA

Figure 4-508. TCMAMEMTAB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																TCMAMEMTAB															
0h																R/W-7654h															

Table 4-536. TCMAMEMTAB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU1		0h	Not Used
15-0	TCMAMEMTAB	R/W	7654h	MSS TCMA memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 4), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 5), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 6), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 7). Corresponding banks need to be selected for MSS TCMA in SHMEMBANKSEL regsiter

4.8.1.41 TCMBMEMTAB Register (Offset = 2C0h) [reset = 7654h]

TCMBMEMTAB is shown in Figure 4-807 and described in Table 4-844.

Return to Summary Table.

Controls ordering of banks in shared memory associated with MSS TCMB

Figure 4-509. TCMBMEMTAB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																TCMBMEMTAB															
0h																R/W-7654h															

Table 4-537. TCMBMEMTAB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU1		0h	Not Used
15-0	TCMBMEMTAB	R/W	7654h	MSS TCMB memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 4), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 5), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 6), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 7). Corresponding banks need to be selected for MSS TCMB in SHMEMBANKSEL regsiter

4.8.1.42 SHMEMBANKSEL3TO0 Register (Offset = 2C8h) [reset = 01010101h]

SHMEMBANKSEL3TO0 is shown in Figure 4-808 and described in Table 4-845.

Return to Summary Table.

Shared memory master allocation.Writing to each 8 bit field indicates the bank allocated to which master. 0x1 : DSS 0x2 : MSS TCMA 0x4 : MSS TCMB 0x8 : OCLA 0x10: BSS TCMA

Figure 4-510. SHMEMBANKSEL3TO0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BANK3								BANK2								BANK1								BANK0							
R/W-1h								R/W-1h								R/W-1h								R/W-1h							

Table 4-538. SHMEMBANKSEL3TO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BANK3	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
23-16	BANK2	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
15-8	BANK1	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
7-0	BANK0	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)

4.8.1.43 SHMEMBANKSEL7TO4 Register (Offset = 2CCh) [reset = 01010101h]

SHMEMBANKSEL7TO4 is shown in Figure 4-809 and described in Table 4-846.

Return to Summary Table.

Shared memory master allocation.Writing to each 8 bit field indicates the bank allocated to which master. 0x1 : DSS 0x2 : MSS TCMA 0x4 : MSS TCMB 0x8 : OCLA 0x10: BSS TCMA

Figure 4-511. SHMEMBANKSEL7TO4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BANK7								BANK6								BANK5								BANK4							
R/W-1h								R/W-1h								R/W-1h								R/W-1h							

Table 4-539. SHMEMBANKSEL7TO4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BANK7	R/W	1h	only valid value is 0x1/0x2/0x4/0x8/0X10 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA,BSS TCMA)
23-16	BANK6	R/W	1h	only valid value is 0x1/0x2/0x4/0x8/0X10 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA,BSS TCMA)
15-8	BANK5	R/W	1h	only valid value is 0x1/0x2/0x4/0x8 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA)
7-0	BANK4	R/W	1h	only valid value is 0x1/0x2/0x4/0x8 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA)

4.8.1.44 PBISTCLKCTL Register (Offset = 2D0h) [reset = 1h]

PBISTCLKCTL is shown in Figure 4-810 and described in Table 4-847.

Return to Summary Table.

PBIST clock control register

Figure 4-512. PBISTCLKCTL Register

31	30	29	28	27	26	25	24
NU							
0h							
23	22	21	20	19	18	17	16
NU							
0h							
15	14	13	12	11	10	9	8
PBIST300MCLKGATE							
R/W-0h							
7	6	5	4	3	2	1	0
PBIST300MCLKSRCSEL				PBIST300MCLKDIV			
R/W-0h				R/W-1h			

Table 4-540. PBISTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Not used
15-8	PBIST300MCLKGATE	R/W	0h	Pre clock divider gate for PBIST300M clock. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA
7-4	PBIST300MCLKSRCSEL	R/W	0h	Select clock source for DSP PBIST Clock source 000 => CPUCLK 001, 100, 101, 111 => RCCLK 010 => 600-MHz PLL divided clock 110 => REFCLK from ANA
3-0	PBIST300MCLKDIV	R/W	1h	Divide value for DSP PBIST source clock selected by field PBIST300MCLKSRCSEL in register CLKSRCSEL. “0000” => div1 “0001” => div2 | | “1111” => div15 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming PBIST300MCLKSRCSEL

4.8.2 MSS_RCM Registers

Table 4-541 lists the MSS_RCM registers. All register offset addresses not listed in Table 4-541 should be considered as reserved locations and the register contents should not be modified.

Table 4-541. MSS_RCM Registers

Offset	Acronym	Register Name	Section
0h	SOFTRST0	SYS_SOFT_RESET0	Section 4.8.2.1
4h	SOFTRST1	SYS_SOFT_RESET1	Section 4.8.2.2
8h	SOFTRST2	SYS_SOFT_RESET2	Section 4.8.2.3
Ch	SOFTRST3	SYS_SOFT_RESET3	Section 4.8.2.4
10h	MCUIFSOFTRST0	MCU_IF_SOFT_RESET0	Section 4.8.2.5
14h	ECUIFSOFTRST0	ECU_IF_SOFT_RESET0	Section 4.8.2.6
18h	CLKDIVCTL0	CLKDIV	Section 4.8.2.7
1Ch	CLKSRCSEL0	CLKSRCSEL	Section 4.8.2.8
20h	CR4CTL	CR4CTL	Section 4.8.2.9
24h	SOFTRST4	SYS_SOFT_RESET4	Section 4.8.2.10
3Ch	CLKGATE	CLK_GATE	Section 4.8.2.11
44h	CLKSRCSEL1	SYS_CLKSRCSEL	Section 4.8.2.12
50h	CLKDIVCTL1	RTI_CLKDIV	Section 4.8.2.13
54h	CURRCLKDIV0	CURR_CLKDIV	Section 4.8.2.14
58h	RTICURRCLKDIV	RTI_CURR_CLKDIV	Section 4.8.2.15
5Ch	MEMINITSTART	MEM_INIT_START	Section 4.8.2.16
60h	CURRCLKDIV1	CURR_CLKDIV1	Section 4.8.2.17
6Ch	MEMINITDONE	MEM_INIT_DONE	Section 4.8.2.18
70h	ECCENMSSGEM	ECCEN_MSSGEM	Section 4.8.2.19
74h	ECCCAPTMSSGEM	ECCCAPT_MSSGEM	Section 4.8.2.20
78h	ECCENBSSGEM	ECCEN_BSSGEM	Section 4.8.2.21
7Ch	ECCCAPTBSSGEM	ECCCAPT_BSSGEM	Section 4.8.2.22
80h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.8.2.23
84h	NSYSPERUSERMODEN	NON_SYS_PERIPH_USERMODEEN	Section 4.8.2.24
88h	SECURERAMMMI	SECURERAMMMI	Section 4.8.2.25
8Ch	SECURERAMECC	SECURERAMECC	Section 4.8.2.26
90h	ESMGATE0	ESMGATE0	Section 4.8.2.27
94h	ESMGATE1	ESMGATE1	Section 4.8.2.28
98h	ESMGATE2	ESMGATE2	Section 4.8.2.29
9Ch	ESMGATE3	ESMGATE3	Section 4.8.2.30
A0h	ESMGATE4	ESMGATE4	Section 4.8.2.31
A8h	MSSECOSPARE		Section 4.8.2.32
ACh	KEY	CFGREG_ACCESS_KEY	Section 4.8.2.33
B0h	DBGACKCTL0	DEBUG ACK ENABLE	Section 4.8.2.34
B4h	DBGACKCTL1	DEBUG ACK ENABLE	Section 4.8.2.35
B8h	SWIRQA	SWIRQ0	Section 4.8.2.36
BCh	SWIRQB	SWIRQ1	Section 4.8.2.37
C0h	MISCCTL0	MISCELLANEOUS_CTL_REG	Section 4.8.2.38
C4h	ATCMERRCAPTCTL	ATCMERRCAPT	Section 4.8.2.39
C8h	B0TCMERRCAPTCTL	B0TCMERRCAPT	Section 4.8.2.40
CCh	B1TCMERRCAPTCTL	B1TCMERRCAPT	Section 4.8.2.41
D0h	SOFTCORERST	SOFT_CORE_RST	Section 4.8.2.42
D4h	WDOGRSTEN	WDOG_RST_EN	Section 4.8.2.43
D8h	RSTCAUSE	MSS_RST_CAUSE	Section 4.8.2.44

Table 4-541. MSS_RCM Registers (continued)

Offset	Acronym	Register Name	Section
DCh	RSTCAUSECLR	MSS_RST_CAUSE_CLR	Section 4.8.2.45
E0h	SPITRIGSRC	SPI_TRIG_SRC	Section 4.8.2.46
E4h	CLKINUSE	MSS_CLK_IN_USE	Section 4.8.2.47
E8h	ECCENMSSBSS	MSS_ECC_EN	Section 4.8.2.48
ECh	ECCCAPTMSSBSS	MSS_ECC_CAPT	Section 4.8.2.49
F0h	CLKDIVCTL2	CLKDIV2	Section 4.8.2.50
F4h	QSPIMCONNECT	QSPI_MCONNECT	Section 4.8.2.51
F8h	QSPISCONNECT	QSPI_SCONNECT	Section 4.8.2.52
FCh	SWIRQC	SW_IRQ2	Section 4.8.2.53

4.8.2.1 SOFTRST0 Register (Offset = 0h) [reset = 0h]

SOFTRST0 is shown in Figure 4-513 and described in Table 4-542.

Return to the Table 4-541.

Figure 4-513. SOFTRST0 Register

31	30	29	28	27	26	25	24
NU							SYSRSTDBGRSTEN
0h							R/W-0h
23	22	21	20	19	18	17	16
SYSRSTCLKGATEEN							
R/W-0h							
15	14	13	12	11	10	9	8
SYSRSTIDLECHKEN							
R/W-0h							
7	6	5	4	3	2	1	0
SYSRST							
0h							

Table 4-542. SOFTRST0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU		0h	
24	SYSRSTDBGRSTEN	R/W	0h	This bit is implemented as multibit in HW. 1 : DBG Reset by writing to CR4 debug register will trigger MSS subsystem reset. 0 : Trigger CR4 only reset
23-16	SYSRSTCLKGATEEN	R/W	0h	0xAD : Gate MSS clock before MSS subsystem reset and ungate after reset assertion. 0x0 : Dont gate clock before reset assertion.
15-8	SYSRSTIDLECHKEN	R/W	0h	0xAD : Reset MSS only after cross subsystem buses are in idle state. I.e : No pending transaction. This is applicable for MSS Wdog reset and SW reset using SYSRST 0X0: Dont wait for cross susbsytem to be idle. - This can hang the bus when the reset happens when there is cross susbsystem transaction going on.
7-0	SYSRST		0h	Write 0xAD to assert a MSS subsystem only reset.Self clearing By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

4.8.2.2 SOFTRST1 Register (Offset = 4h) [reset = 0h]

SOFTRST1 is shown in Figure 4-514 and described in Table 4-543.

Return to the Table 4-541.

Figure 4-514. SOFTRST1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMARST								DCCBRST							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCCARST								CR4SYSRST							
R/W-0h								0h							

Table 4-543. SOFTRST1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DMARST	R/W	0h	Write 0xAD to assert a DMA only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-16	DCCBRST	R/W	0h	Write 0xAD to assert a DCCB only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
15-8	DCCARST	R/W	0h	Write 0xAD to assert a DCCA only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
7-0	CR4SYSRST		0h	Write 0xAD to assert a MSS CR4 only reset.Self clearing By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

4.8.2.3 SOFTRST2 Register (Offset = 8h) [reset = 0h]

SOFTRST2 is shown in Figure 4-515 and described in Table 4-544.

Return to the Table 4-541.

Figure 4-515. SOFTRST2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIMRST								ESMRST								RTIRST								UARTARST							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 4-544. SOFTRST2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VIMRST	R/W	0h	Write 0xAD to assert a VIM only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-16	ESMRST	R/W	0h	Write 0xAD to assert a ESM only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
15-8	RTIRST	R/W	0h	Write 0xAD to assert a RTI only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
7-0	UARTARST	R/W	0h	Write 0xAD to assert a UARTA/SCIA only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

4.8.2.4 SOFTRST3 Register (Offset = Ch) [reset = 0h]

SOFTRST3 is shown in Figure 4-516 and described in Table 4-545.

Return to the Table 4-541.

Figure 4-516. SOFTRST3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRCRST								SCRRST								CRCRST								WDTRST							
R/W-0h								0h								R/W-0h								R/W-0h							

Table 4-545. SOFTRST3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FRCRST	R/W	0h	Write 0xAD to assert a FRC only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-16	SCRRST		0h	Reserved. Altering the reset value can impact Chip functionality.
15-8	CRCRST	R/W	0h	Write 0xAD to assert a CRC only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
7-0	WDTRST	R/W	0h	Write 0xAD to assert a WDT (RTI used for watch dog) only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

4.8.2.5 MCUIFSOFTRST0 Register (Offset = 10h) [reset = 0h]

MCUIFSOFTRST0 is shown in Figure 4-517 and described in Table 4-546.

Return to the Table 4-541.

Figure 4-517. MCUIFSOFTRST0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIORST								SPIBRST								SPIARST								QSPIRST							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 4-546. MCUIFSOFTRST0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIORST	R/W	0h	Write 0xAD to assert a GIO only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-16	SPIBRST	R/W	0h	Write 0xAD to assert a SPIB only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
15-8	SPIARST	R/W	0h	Write 0xAD to assert a SPIA only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
7-0	QSPIRST	R/W	0h	Write 0xAD to assert a QSPI only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

4.8.2.6 ECUIFSOFTRST0 Register (Offset = 14h) [reset = 0h]

ECUIFSOFTRST0 is shown in Figure 4-518 and described in Table 4-547.

Return to the Table 4-541.

Figure 4-518. ECUIFSOFTRST0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UARTBRST								CANFDRST							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CRST								FDCAN2RST							
R/W-0h								R/W-0h							

Table 4-547. ECUIFSOFTRST0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	UARTBRST	R/W	0h	Write 0xAD to assert a UARTB/SCIB only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-16	CANFDRST	R/W	0h	Write 0xAD to assert a CANFD only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA. PS: Not applicable for AR12XX/AR14XX
15-8	I2CRST	R/W	0h	Write 0xAD to assert a I2C only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
7-0	FDCAN2RST	R/W	0h	Write 0xAD to assert a FDCAN2 only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

4.8.2.7 CLKDIVCTL0 Register (Offset = 18h) [reset = 0h]

CLKDIVCTL0 is shown in Figure 4-519 and described in Table 4-548.

Return to the Table 4-541.

Figure 4-519. CLKDIVCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCANCLKDIV								FDCAN2CLKDIV							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCLKDIV								PCRCLKDIV							
R/W-0h								R/W-0h							

Table 4-548. CLKDIVCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FDCANCLKDIV	R/W	0h	Divide value for FDCAN source clock selected by field FDCANCLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
23-16	FDCAN2CLKDIV	R/W	0h	Divide value for FDCAN2 source clock selected by field FDDCAN2CLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
15-8	VCLKCLKDIV	R/W	0h	Divide value for MSS subsystem source clock selected by field VCLKCLKSRCSEL in register CLKSRCSEL1 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
7-0	PCRCLKDIV	R/W	0h	Reserved. Altering the reset value can impact Chip functionality.

4.8.2.8 CLKSRCSEL0 Register (Offset = 1Ch) [reset = 0h]

CLKSRCSEL0 is shown in Figure 4-814 and described in Table 4-853.

Return to the Table 4-541.

Figure 4-520. CLKSRCSEL0 Register

31	30	29	28	27	26	25	24
RTIDCLKSRCSEL				RTICCLKSRCSEL			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
NU3				QSPICLKSRCSEL			
0h				R/W-0h			
15	14	13	12	11	10	9	8
NU2				FDCANCLKSRCSEL			
0h				R/W-0h			
7	6	5	4	3	2	1	0
NU1				FDCAN2CLKSRCSEL			
0h				R/W-0h			

Table 4-549. CLKSRCSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RTIDCLKSRCSEL	R/W	0h	Select clock source for RTID baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
27-24	RTICCLKSRCSEL	R/W	0h	Select clock source for RTIC baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
23-20	NU3		0h	
19-16	QSPICLKSRCSEL	R/W	0h	Select clock source for QSPI baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
15-12	NU2		0h	
11-8	FDCANCLKSRCSEL	R/W	0h	Select clock source for FDCANCLKSRCSEL 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA 111 => RCCLK
7-4	NU1		0h	
3-0	FDCAN2CLKSRCSEL	R/W	0h	Select clock source for FDCAN2 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK

4.8.2.9 CR4CTL Register (Offset = 20h) [reset = 0h]

CR4CTL is shown in Figure 4-521 and described in Table 4-550.

Return to the Table 4-541.

Figure 4-521. CR4CTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU								MEMSWAPWAIT							
0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR4MEMSWAP								BIGEND							
R/W-0h								R/W-0h							

Table 4-550. CR4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU		0h	Reserved
23-16	MEMSWAPWAIT	R/W	0h	When CR4MEMSWAP is 0xAD : Write 0xAD to this field enable the CR4MEMSWAP only after a CR4 reset – either by writing to CR4SYSRST or by writing to PRCR register in CR4 debug space.
15-8	CR4MEMSWAP	R/W	0h	Write 0xAD will map the MSS CR4 0x0000_0000 to MSS CR4 TCMA RAM start address.
7-0	BIGEND	R/W	0h	Reserved. Modification of reset value can effect the functionality.

4.8.2.10 SOFTRST4 Register (Offset = 24h) [reset = 0h]

SOFTRST4 is shown in Figure 4-522 and described in Table 4-551.

Return to the Table 4-541.

Figure 4-522. SOFTRST4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																RTIDRST								RTICRST							
0h																R/W-0h								R/W-0h							

Table 4-551. SOFTRST4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	
15-8	RTIDRST	R/W	0h	Write 0xAD to assert a RTID only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
7-0	RTICRST	R/W	0h	Write 0xAD to assert a RTIC only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

4.8.2.11 CLKGATE Register (Offset = 3Ch) [reset = 400h]

CLKGATE is shown in Figure 4-816 and described in Table 4-855.

Return to the Table 4-541.

Figure 4-523. CLKGATE Register

31	30	29	28	27	26	25	24
NU							
0h							
23	22	21	20	19	18	17	16
NU							
0h							
15	14	13	12	11	10	9	8
NU					FRCCLKGATE	SYSVCLKGATE	ECUVCLKGATE
0h					R/W-1h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RTIDCLKGATE	RTICCLKGATE	FDCANCLKGATE	FDCAN2CLKGATE	QSPICLKGATE	PCRVCLKGATE	SCRVCLKGATE	CR4VCLKGATE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-552. CLKGATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU		0h	
10	FRCCLKGATE	R/W	1h	Pre clock divider gate for FRC clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
9	SYSVCLKGATE	R/W	0h	Not used.Reserved
8	ECUVCLKGATE	R/W	0h	Not used.Reserved
7	RTIDCLKGATE	R/W	0h	Pre clock divider gate for RTID clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
6	RTICCLKGATE	R/W	0h	Pre clock divider gate for RTIC clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
5	FDCANCLKGATE	R/W	0h	Pre clock divider gate for FDCAN clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
4	FDCAN2CLKGATE	R/W	0h	Pre clock divider gate for FDCAN2 clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
3	QSPICLKGATE	R/W	0h	Pre clock divider gate for QSPI clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
2	PCRVCLKGATE	R/W	0h	Not used.Reserved
1	SCRVCLKGATE	R/W	0h	Not used.Reserved
0	CR4VCLKGATE	R/W	0h	Not used.Reserved

4.8.2.12 CLKSRCSEL1 Register (Offset = 44h) [reset = 0h]

CLKSRCSEL1 is shown in Figure 4-524 and described in Table 4-553.

Return to the Table 4-541.

Figure 4-524. CLKSRCSEL1 Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1				FRCCLKSRCSEL			
0h				R/W-0h			
7	6	5	4	3	2	1	0
NU0				VCLKCLKSRCSEL			
0h				R/W-0h			

Table 4-553. CLKSRCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU1		0h	
11-8	FRCCLKSRCSEL	R/W	0h	Select clock source for FRC 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
7-4	NU0		0h	
3-0	VCLKCLKSRCSEL	R/W	0h	Select clock source for MSS subsystem VCLK 000 =>CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => RCCLK (10Mhz) 101 => RCCLK (10Mhz) 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK(10Mhz)

4.8.2.13 CLKDIVCTL1 Register (Offset = 50h) [reset = 0h]

CLKDIVCTL1 is shown in Figure 4-525 and described in Table 4-554.

Return to the Table 4-541.

Figure 4-525. CLKDIVCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU								FRCCLKDIV							
0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTIDCLKDIV								RTICCLKDIV							
R/W-0h								R/W-0h							

Table 4-554. CLKDIVCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU		0h	
23-16	FRCCLKDIV	R/W	0h	Divide value for FRC source clock selected by field FRCCLKSRCSEL in register CLKSRCSEL1 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
15-8	RTIDCLKDIV	R/W	0h	Select clock source for RTID 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
7-0	RTICCLKDIV	R/W	0h	Select clock source for RTIC 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK

4.8.2.14 CURRCLKDIV0 Register (Offset = 54h) [reset = 0h]

CURRCLKDIV0 is shown in Figure 4-526 and described in Table 4-555.

Return to the Table 4-541.

Figure 4-526. CURRCLKDIV0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCANCURRCLKDIV								FDCAN2CURRCLKDIV							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCURRCLKDIV								PCRCURRCLKDIV							
R-0h								R-0h							

Table 4-555. CURRCLKDIV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FDCANCURRCLKDIV	R	0h	Returns Current divide value of FDCAN baud clock divider.
23-16	FDCAN2CURRCLKDIV	R	0h	Returns Current divide value of FDCAN2 baud clock divider.
15-8	VCLKCURRCLKDIV	R	0h	Returns Current divide value of VCLK divider.
7-0	PCRCURRCLKDIV	R	0h	Reserved

4.8.2.15 RTICURRCLKDIV Register (Offset = 58h) [reset = 0h]

RTICURRCLKDIV is shown in Figure 4-527 and described in Table 4-556.

Return to the Table 4-541.

Figure 4-527. RTICURRCLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU								FRCCURRCLKDIV							
0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTIDCURRCLKDIV								RTICCURRCLKDIV							
R-0h								R-0h							

Table 4-556. RTICURRCLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU		0h	
23-16	FRCCURRCLKDIV	R	0h	Returns Current divide value of FRC clock divider.
15-8	RTIDCURRCLKDIV	R	0h	Returns Current divide value of RTID divider.
7-0	RTICCURRCLKDIV	R	0h	Returns Current divide value of RTIC divider.

4.8.2.16 MEMINITSTART Register (Offset = 5Ch) [reset = 0h]

MEMINITSTART is shown in Figure 4-528 and described in Table 4-557.

Return to the Table 4-541.

Figure 4-528. MEMINITSTART Register

31	30	29	28	27	26	25	24
MEMINITKEY							
R/W-0h							
23	22	21	20	19	18	17	16
NU3						BSSMBOX4GEMMEM	MSSMBOX4GEMMEM
0h						0h	0h
15	14	13	12	11	10	9	8
GEMMBOX4MSSMEM	GEMMBOX4BSSMEM	FTUMEM	NU1		NU2	DMA2MEM	BSSMBOX4MSSMEM
0h	0h	0h	0h		0h	0h	0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	NU	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-557. MEMINITSTART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MEMINITKEY	R/W	0h	Memory hardware initialization global enable key. Write 0XAD to enable MEMINIT.
23-18	NU3		0h	
17	BSSMBOX4GEMMEM		0h	Writing '1' will trigger DSS- BSS mailbox initialization.Self clearing
16	MSSMBOX4GEMMEM		0h	Writing '1' will trigger DSS- MSS mailbox initialization.Self clearing
15	GEMMBOX4MSSMEM		0h	Writing '1' will trigger DSS- MSS mailbox initialization.Self clearing
14	GEMMBOX4BSSMEM		0h	Writing '1' will trigger DSS- BSS mailbox initialization.Self clearing
13	FTUMEM		0h	Not used.Reserved
12-11	NU1		0h	
10	NU2		0h	
9	DMA2MEM		0h	Writing '1' will trigger DMA2 memory initialization.Self clearing
8	BSSMBOX4MSSMEM		0h	Writing '1' will trigger MSS- BSS mailbox initialization.Self clearing
7	MSSMBOX4BSSMEM		0h	Writing '1' will trigger MSS- BSS mailbox initialization.Self clearing
6	NU		0h	
5	SPIBMEM		0h	Writing '1' will trigger SPIB memory initialization.Self clearing
4	SPIAMEM		0h	Writing '1' will trigger SPIA memory initialization.Self clearing
3	VIMMEM		0h	Writing '1' will trigger VIM memory initialization.Self clearing
2	DMAMEM		0h	Writing '1' will trigger DMA memory initialization.Self clearing
1	CR4TCMBMEM		0h	Writing '1' will trigger MSS TCMB memory initialization.Self clearing
0	CR4TCMAMEM		0h	Writing '1' will trigger MSS TCMA memory initialization.Self clearing

4.8.2.17 CURRCLKDIV1 Register (Offset = 60h) [reset = 0h]

CURRCLKDIV1 is shown in Figure 4-529 and described in Table 4-558.

Return to the Table 4-541.

Figure 4-529. CURRCLKDIV1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MISCCLKCURRCLKDIV															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCLKCURRCLKDIV								QSPICURRCLKDIV							
R-0h								R-0h							

Table 4-558. CURRCLKDIV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	MISCCLKCURRCLKDIV	R	0h	Reserved for future use
7-0	QSPICURRCLKDIV	R	0h	Returns Current divide value of QSPI_CLK divider.

4.8.2.18 MEMINITDONE Register (Offset = 6Ch) [reset = 0h]

MEMINITDONE is shown in Figure 4-530 and described in Table 4-559.

Return to the Table 4-541.

Figure 4-530. MEMINITDONE Register

31	30	29	28	27	26	25	24
NU2							
0h							
23	22	21	20	19	18	17	16
NU2						BSSMBOX4GEMMEM	MSSMBOX4GEMMEM
0h						R-0h	R-0h
15	14	13	12	11	10	9	8
GEMMBOX4MSSMEM	GEMMBOX4BSSMEM	FTUMEM	NU1		NU3	DMA2MEM	BSSMBOX4MSSMEM
R-0h	R-0h	R-0h	0h		0h	R-0h	R-0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	NU	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
R-0h	0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 4-559. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	NU2		0h	
17	BSSMBOX4GEMMEM	R	0h	Memory Initialization done status for DSS- BSS mailbox
16	MSSMBOX4GEMMEM	R	0h	Memory Initialization done status for DSS- MSS mailbox
15	GEMMBOX4MSSMEM	R	0h	Memory Initialization done status for DSS- MSS mailbox
14	GEMMBOX4BSSMEM	R	0h	Memory Initialization done status for DSS- BSS mailbox
13	FTUMEM	R	0h	Not used.Reserved
12-11	NU1		0h	
10	NU3		0h	
9	DMA2MEM	R	0h	Memory Initialization done status for MSS DMA2 memory
8	BSSMBOX4MSSMEM	R	0h	Memory Initialization done status for MSS- BSS mailbox
7	MSSMBOX4BSSMEM	R	0h	Memory Initialization done status for MSS- BSS mailbox
6	NU		0h	
5	SPIBMEM	R	0h	Memory Initialization done status for MSS SPIB memory
4	SPIAMEM	R	0h	Memory Initialization done status for MSS SPIA memory
3	VIMMEM	R	0h	Memory Initialization done status for MSS VIM memory
2	DMAMEM	R	0h	Memory Initialization done status for MSS DMA memory
1	CR4TCMBMEM	R	0h	Memory Initialization done status for MSS TCMB memory
0	CR4TCMAMEM	R	0h	Memory Initialization done status for MSS TCMA memory

4.8.2.19 ECCENMSSGEM Register (Offset = 70h) [reset = 0h]

ECCENMSSGEM is shown in Figure 4-531 and described in Table 4-560.

Return to the Table 4-541.

Figure 4-531. ECCENMSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENMSSGEM																															
R/W-0h																															

Table 4-560. ECCENMSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENMSSGEM	R/W	0h	7:0 : Writing 0xAD will enable ECC for MSS- DSS mailbox 15:8: Writing 0xAD will enable ECC for MSS- DSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in MSS mailbox for GEM 21:19: Write 3'b111 to clear the Address captured because of ECC error. In GEM mailbox for MSS

4.8.2.20 ECCCAPTMSSGEM Register (Offset = 74h) [reset = 0h]

ECCCAPTMSSGEM is shown in Figure 4-532 and described in Table 4-561.

Return to the Table 4-541.

Figure 4-532. ECCCAPTMSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTMSSGEM																															
R-0h																															

Table 4-561. ECCCAPTMSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTMSSGEM	R	0h	7:0 : mss_mbox4gem_ecc_fault_address 14:8 : mss_mbox4gem_repaired_bit 23:16 : gem_mbox4mss_ecc_fault_address 30:24 : gem_mbox4mss_repaired_bit

4.8.2.21 ECCENBSSGEM Register (Offset = 78h) [reset = 0h]

ECCENBSSGEM is shown in Figure 4-533 and described in Table 4-562.

Return to the Table 4-541.

Figure 4-533. ECCENBSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENBSSGEM																															
R/W-0h																															

Table 4-562. ECCENBSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENBSSGEM	R/W	0h	7:0 : Writing 0xAD will enable ECC for DSS- BSS mailbox 15:8: Writing 0xAD will enable ECC for DSS- BSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in GEM mailbox for BSS 21:19: Write 3'b111 to clear the Address captured because of ECC error in BSS mailbox for GEM

4.8.2.22 ECCCAPTBSSGEM Register (Offset = 7Ch) [reset = 0h]

ECCCAPTBSSGEM is shown in Figure 4-534 and described in Table 4-563.

Return to the Table 4-541.

Figure 4-534. ECCCAPTBSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTBSSGEM																															
R-0h																															

Table 4-563. ECCCAPTBSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTBSSGEM	R	0h	7:0 : bss_mbox4gem_ecc_fault_address 14:8 : bss_mbox4gem_repaired_bit 23:16 : gem_mbox4bss_ecc_fault_address 30:24 : gem_mbox4bss_repaired_bit

4.8.2.23 USERMODEEN Register (Offset = 80h) [reset = 0h]

USERMODEEN is shown in Figure 4-535 and described in Table 4-564.

Return to the Table 4-541.

Figure 4-535. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-564. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS RCM space.

4.8.2.24 NSYSPERUSERMODEN Register (Offset = 84h) [reset = 0h]

NSYSPERUSERMODEN is shown in Figure 4-536 and described in Table 4-565.

Return to the Table 4-541.

Figure 4-536. NSYSPERUSERMODEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSYSPERUSERMODEN																															
R/W-0h																															

Table 4-565. NSYSPERUSERMODEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NSYSPERUSERMODEN	R/W	0h	2:0 : Write 3'b111 to enable user mode access to SPIA 5:3 : Write 3'b111 to enable user mode access to SPIB 10:8 : Write 3'b111 to enable user mode access to GIO 13:11 : Write 3'b111 to enable user mode access to QSPI 18:16 : Write 3'b111 to enable user mode access to SCIA 21:19 : Write 3'b111 to enable user mode access to SCIB 26:24 : Write 3'b111 to enable user mode access to DCAN

4.8.2.25 SECURERAMMMI Register (Offset = 88h) [reset = 0h]

SECURERAMMMI is shown in Figure 4-537 and described in Table 4-566.

Return to the Table 4-541.

Figure 4-537. SECURERAMMMI Register

31	30	29	28	27	26	25	24
NU2							
0h							
23	22	21	20	19	18	17	16
NU2							SECURERAMINITDONE
0h							R-0h
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
NU1							SECURERAMINIT
0h							0h

Table 4-566. SECURERAMMMI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU2		0h	
16	SECURERAMINITDONE	R	0h	Memory Initialization done status for Secure Key RAM
15-1	NU1		0h	
0	SECURERAMINIT		0h	Writing '1' will trigger Secure Key RAM memory initialization.Self clearing

4.8.2.26 SECURERAMECC Register (Offset = 8Ch) [reset = 0h]

SECURERAMECC is shown in Figure 4-538 and described in Table 4-567.

Return to the Table 4-541.

Figure 4-538. SECURERAMECC Register

31	30	29	28	27	26	25	24
SECURERAMBIT							
R-0h							
23	22	21	20	19	18	17	16
SECURERAMADDR							
R-0h							
15	14	13	12	11	10	9	8
NU					SECURERAMECCCLR		
0h					R/W-0h		
7	6	5	4	3	2	1	0
SECURERAMECCEN							
R/W-0h							

Table 4-567. SECURERAMECC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SECURERAMBIT	R	0h	Secure key RAM repaired bit
23-16	SECURERAMADDR	R	0h	Secure Key RAM_ecc_fault_address
15-11	NU		0h	
10-8	SECURERAMECCCLR	R/W	0h	10:8 : Write 3'b111 to clear the Address captured because of ECC error
7-0	SECURERAMECCEN	R/W	0h	7:0 : Writing 0xAD will enable ECC for Secure key RAM

4.8.2.27 ESMGATE0 Register (Offset = 90h) [reset = 07070700h]

ESMGATE0 is shown in Figure 4-539 and described in Table 4-568.

Return to the Table 4-541.

Figure 4-539. ESMGATE0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE0																															
R/W-07070700h																															

Table 4-568. ESMGATE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE0	R/W	07070700h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 0 [7:4] : Gate ESM group2 line 1 [11:8] : Gate ESM group2 line 2 [15:12] : Gate ESM group2 line 3 [19:16] : Gate ESM group2 line 4 [23:20] : Gate ESM group2 line 5 [27:24] : Gate ESM group2 line 6 [31:28] : Gate ESM group2 line 7 Static register setting.Should not be changed on the fly

4.8.2.28 ESMGATE1 Register (Offset = 94h) [reset = 7h]

ESMGATE1 is shown in Figure 4-540 and described in Table 4-569.

Return to the Table 4-541.

Figure 4-540. ESMGATE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE1																															
R/W-7h																															

Table 4-569. ESMGATE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE1	R/W	7h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 8 [7:4] : Gate ESM group2 line 9 [11:8] : Gate ESM group2 line 10 [15:12] : Gate ESM group2 line 11 [19:16] : Gate ESM group2 line 12 [23:20] : Gate ESM group2 line 13 [27:24] : Gate ESM group2 line 14 [31:28] : Gate ESM group2 line 15 Static register setting.Should not be changed on the fly

4.8.2.29 ESMGATE2 Register (Offset = 98h) [reset = 7h]

ESMGATE2 is shown in Figure 4-541 and described in Table 4-570.

Return to the Table 4-541.

Figure 4-541. ESMGATE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE2																															
R/W-7h																															

Table 4-570. ESMGATE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE2	R/W	7h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23 Static register setting.Should not be changed on the fly

4.8.2.30 ESMGATE3 Register (Offset = 9Ch) [reset = 3Fh]

ESMGATE3 is shown in Figure 4-542 and described in Table 4-571.

Return to the Table 4-541.

Figure 4-542. ESMGATE3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE3																															
R/W-3Fh																															

Table 4-571. ESMGATE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE3	R/W	3Fh	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23 Static register setting.Should not be changed on the fly

4.8.2.31 ESMGATE4 Register (Offset = A0h) [reset = 0h]

ESMGATE4 is shown in Figure 4-543 and described in Table 4-572.

Return to the Table 4-541.

Figure 4-543. ESMGATE4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE4																															
R/W-0h																															

Table 4-572. ESMGATE4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE4	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group3 line 0 [7:4] : Gate ESM group3 line 1 [11:8] : Gate ESM group3 line 2 [15:12] : Gate ESM group3 line 3 [19:16] : Gate ESM group3 line 4 [23:20] : Gate ESM group3 line 5 [27:24] : Gate ESM group3 line 6 [31:28] : Gate ESM group3 line 7 Static register setting.Should not be changed on the fly

4.8.2.32 MSSECOSPARE Register (Offset = A8h) [reset = 0h]

MSSECOSPARE is shown in Figure 4-544 and described in Table 4-573.

Return to the Table 4-541.

Figure 4-544. MSSECOSPARE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSSECOSPARE																															
R/W-0h																															

Table 4-573. MSSECOSPARE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSSECOSPARE	R/W	0h	[0] : cfg_DBGRST12xx_SELECT. '1' selects DBGRST to MSS CR4 same as AR12XX i.e both debug logic and CR4 Core gets reset with DBGRST '0' DBGRST resets only CR4 Core

4.8.2.33 KEY Register (Offset = ACh) [reset = 83E783E7h]

KEY is shown in Figure 4-545 and described in Table 4-574.

Return to the Table 4-541.

Figure 4-545. KEY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY																															
R/W-83E783E7h																															

Table 4-574. KEY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY	R/W	83E783E7h	Kicker Register. The value 83E7_83E7h must be written as part of the process to unlock the CPU write access to the MSS RCM registers

4.8.2.34 DBGACKCTL0 Register (Offset = B0h) [reset = FFFFFFFFh]

DBGACKCTL0 is shown in Figure 4-546 and described in Table 4-575.

Return to the Table 4-541.

Figure 4-546. DBGACKCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBGACKCTL0																															
R/W-FFFFFFFFh																															

Table 4-575. DBGACKCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBGACKCTL0	R/W	FFFFFFFFh	When MSS CR4 enters debug halt state ie, when DBGACK is asserted ,the peripherals (that supports the feature) can also be put in suspend mode .This feature can be enabled or disabled for the different peripherals using this register [29:27] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module DCCA [26:24] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module DCCB [21:19] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module DCAN [18:16] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module I2C [13:11] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module MCRC [10:8] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module DMA [5:3] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module FRC [2:0] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module WDT

4.8.2.35 DBGACKCTL1 Register (Offset = B4h) [reset = 0h]

DBGACKCTL1 is shown in Figure 4-547 and described in Table 4-576.

Return to the Table 4-541.

Figure 4-547. DBGACKCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBGACKCTL1																															
R/W-0h																															

Table 4-576. DBGACKCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBGACKCTL1	R/W	0h	When MSS CR4 enters debug halt state ie, when DBGACK is asserted ,the peripherals (that supports the feature) can also be put in suspend mode .This feature can be enabled or disabled for the different peripherals using this register [26:24] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module CCMR4 [21:19] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module CCCA [18:16] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module CCCB [13:11] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module UARTA [10:8] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module UARTB [5:3] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module RTI [2:0] :Writing 3'b111 will enable supspend operation when MSS CR4 enters into debug halt state for module BSS

4.8.2.36 SWIRQA Register (Offset = B8h) [reset = 0h]

SWIRQA is shown in Figure 4-548 and described in Table 4-577.

Return to the Table 4-541.

Figure 4-548. SWIRQA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ1								SWIRQ1DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ0								SWIRQ0DAT							
R/W-0h								R/W-0h							

Table 4-577. SWIRQA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ1	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ1DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ0	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ0DAT	R/W	0h	Not Used.Reserved

4.8.2.37 SWIRQB Register (Offset = BCh) [reset = 0h]

SWIRQB is shown in Figure 4-549 and described in Table 4-578.

Return to the Table 4-541.

Figure 4-549. SWIRQB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ3								SWIRQ3DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ2								SWIRQ2DAT							
R/W-0h								R/W-0h							

Table 4-578. SWIRQB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ3	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ3DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ2	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ2DAT	R/W	0h	Not Used.Reserved

4.8.2.38 MISCCTL0 Register (Offset = C0h) [reset = ADADh]

MISCCTL0 is shown in Figure 4-550 and described in Table 4-579.

Return to the Table 4-541.

Figure 4-550. MISCCTL0 Register

31	30	29	28	27	26	25	24
NU							TCMB1EZDIS
0h							R/W-0h
23	22	21	20	19	18	17	16
TCMB1EZDIS			TCMB0EZDIS			TCMAEZDIS	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
SPIBSYNC2SEN							
R/W-ADh							
7	6	5	4	3	2	1	0
SPIASYNC2SEN							
R/W-ADh							

Table 4-579. MISCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU		0h	
24-22	TCMB1EZDIS	R/W	0h	Write 111 to make TCMB1 EZ to '1' regardless of Functional value.
21-19	TCMB0EZDIS	R/W	0h	Write 111 to make TCMB0 EZ to '1' regardless of Functional value.
18-16	TCMAEZDIS	R/W	0h	Write 111 to make TCMA EZ to '1' regardless of Functional value.
15-8	SPIBSYNC2SEN	R/W	ADh	Write 0XAD to enable latching of Baud clock using normal FF else using sync1s FF
7-0	SPIASYNC2SEN	R/W	ADh	Write 0XAD to enable latching of Baud clock using normal FF else using sync1s FF

4.8.2.39 ATCMERRCAPTCTL Register (Offset = C4h) [reset = 0h]

ATCMERRCAPTCTL is shown in Figure 4-551 and described in Table 4-580.

Return to the Table 4-541.

Figure 4-551. ATCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRATCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRATCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRATCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		ATCFORCEERR			ERRATCADDRCLR		
0h		R/W-0h			0h		

Table 4-580. ATCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRATCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	ATCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRATCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.8.2.40 B0TCMERRCAPTCTL Register (Offset = C8h) [reset = 0h]

B0TCMERRCAPTCTL is shown in Figure 4-552 and described in Table 4-581.

Return to the Table 4-541.

Figure 4-552. B0TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB0TCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRB0TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB0TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B0TCFORCEERR			ERRB0TCADDRCLR		
0h		R/W-0h			0h		

Table 4-581. B0TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRB0TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	B0TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB0TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.8.2.41 B1TCMERRCAPTCTL Register (Offset = CCh) [reset = 0h]

B1TCMERRCAPTCTL is shown in Figure 4-553 and described in Table 4-582.

Return to the Table 4-541.

Figure 4-553. B1TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB1TCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRB1TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB1TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B1TCFORCEERR			ERRB1TCADDRCLR		
0h		R/W-0h			0h		

Table 4-582. B1TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRB1TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	B1TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB1TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.8.2.42 SOFTCORERST Register (Offset = D0h) [reset = 000F0F00h]

SOFTCORERST is shown in Figure 4-554 and described in Table 4-583.

Return to the Table 4-541.

Figure 4-554. SOFTCORERST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RST_WFICHECKEN								RSTASSRTDLY							
R/W-0h								R/W-Fh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSTTOASSRTDLY								SOFTCORERST							
R/W-Fh								0h							

Table 4-583. SOFTCORERST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RST_WFICHECKEN	R/W	0h	0xAD : When CR4SYSRST is set, Before asserting the reset to CR4 wait for CR4 to enter WFI. 0X0: Dont wait for WFI
23-16	RSTASSRTDLY	R/W	Fh	Assert below reset for programmed number of clock cycles : 1. CR4 only reset because of either by writing to CR4SYSRST or by writing to PRCR register in CR4 debug space. 2. Subsystem reset because of either writing to SYSRST register or MSS wdog expiry.
15-8	RSTTOASSRTDLY	R/W	Fh	Wait for programmed number of clock cycle before reset is asserted to CR4.
7-0	SOFTCORERST		0h	Reserved

4.8.2.43 WDOGRSTEN Register (Offset = D4h) [reset = 0h]

WDOGRSTEN is shown in Figure 4-555 and described in Table 4-584.

Return to the Table 4-541.

Figure 4-555. WDOGRSTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR4WDOGRSTEN								SYSWDOGRSTEN							
R/W-0h								R/W-0h							

Table 4-584. WDOGRSTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	
15-8	CR4WDOGRSTEN	R/W	0h	Reserved. Altering the reset value can impact Chip functionality.
7-0	SYSWDOGRSTEN	R/W	0h	Enable MSS Wdog expiry to cause only MSS subsystem reset.

4.8.2.44 RSTCAUSE Register (Offset = D8h) [reset = 0h]

RSTCAUSE is shown in Figure 4-556 and described in Table 4-585.

Return to the Table 4-541.

Figure 4-556. RSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								RSTCAUSE							
0h																								R-0h							

Table 4-585. RSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	RSTCAUSE	R	0h	Gives cause of MSS reset 0000_1001 : System out of NRESET 0000_1000 : Warm Reset 0000_0010 : MSS only Wdog Reset. 0000_0100 : MSS subsystem reset because of Software trigger. 0010_0000 : CR4 reset because of Software trigger. 0001_0000 : STC reset 0100_0000 : CR4 reset because of writing to PRCR register in CR4 debug space. 1000_0000 : CR4 only Wdog reset Note: Because the ROM Bootloader issues a soft reset to pass control to the application (or secondary bootloader), this register always reads 0x20. The reset value is stored by the ROM Bootloader into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.8.2.45 RSTCAUSECLR Register (Offset = DCh) [reset = 0h]

RSTCAUSECLR is shown in Figure 4-557 and described in Table 4-586.

Return to the Table 4-541.

Figure 4-557. RSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								RSTCAUSECLR							
0h								0h							

Table 4-586. RSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	
7-0	RSTCAUSECLR		0h	Write 0xAD to clear RSTCAUSE.Self clearing

4.8.2.46 SPITRIGSRC Register (Offset = E0h) [reset = 0h]

SPITRIGSRC is shown in Figure 4-558 and described in Table 4-587.

Return to the Table 4-541.

Figure 4-558. SPITRIGSRC Register

31	30	29	28	27	26	25	24
NU3					SPIBTRIG		
0h					R/W-0h		
23	22	21	20	19	18	17	16
SPIBTRIG							
R/W-0h							
15	14	13	12	11	10	9	8
NU2							SPIATRIG1
0h							R/W-0h
7	6	5	4	3	2	1	0
NU1							SPIATRIG0
0h							R/W-0h

Table 4-587. SPITRIGSRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU3		0h	
26-16	SPIBTRIG	R/W	0h	[20:16] --> Trigger sources for MIBSPIB
15-9	NU2		0h	
8	SPIATRIG1	R/W	0h	1st bit ofTRG_SRC input of SPIA.
7-1	NU1		0h	
0	SPIATRIG0	R/W	0h	0th bit ofTRG_SRC input of SPIA.

4.8.2.47 CLKINUSE Register (Offset = E4h) [reset = 0h]

CLKINUSE is shown in Figure 4-847 and described in Table 4-886.

Return to the Table 4-541.

Figure 4-559. CLKINUSE Register

31	30	29	28	27	26	25	24
NU				FRCCLKINUSE			
0h				R-0h			
23	22	21	20	19	18	17	16
RTIDCLKINUSE				RTICCLKINUSE			
R-0h				R-0h			
15	14	13	12	11	10	9	8
QSPICLKINUSE				FDCAN2CLKINUSE			
R-0h				R-0h			
7	6	5	4	3	2	1	0
FDCANCLKINUSE				VCLKINUSE			
R-0h				R-0h			

Table 4-588. CLKINUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU		0h	
27-24	FRCCLKINUSE	R	0h	Current Clock Source Select Mux value for FRC CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
23-20	RTIDCLKINUSE	R	0h	Current Clock Source Select Mux value for RTID 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
19-16	RTICCLKINUSE	R	0h	Current Clock Source Select Mux value for RTIC 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
15-12	QSPICLKINUSE	R	0h	Current Clock Source Select Mux value for QSPI CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
11-8	FDCAN2CLKINUSE	R	0h	Current Clock Source Select Mux value for FDCAN2 CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
7-4	FDCANCLKINUSE	R	0h	Current Clock Source Select Mux value for FDCAN CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
3-0	VCLKINUSE	R	0h	Current Clock Source Select Mux value for VCLK 000 => CPUCLK (40Mhz or 50 Mhz or 80Mh or 100Mh) 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 101 => RCCLK (10Mhz) 110 => REFCLK 111 => RCCLK (10Mhz)

4.8.2.48 ECCENMSSBSS Register (Offset = E8h) [reset = 0h]

ECCENMSSBSS is shown in Figure 4-560 and described in Table 4-589.

Return to the Table 4-541.

Figure 4-560. ECCENMSSBSS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENMSSBSS																															
R/W-0h																															

Table 4-589. ECCENMSSBSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENMSSBSS	R/W	0h	7:0 : Writing 0xAD will enable ECC for MSS – BSS mailbox 15:8: Writing 0xAD will enable ECC for MSS – BSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in MSS mailbox for BSS 21:19: Write 3'b111 to clear the Address captured because of ECC error. In BSS mailbox for MSS

4.8.2.49 ECCCAPTMSSBSS Register (Offset = ECh) [reset = 0h]

ECCCAPTMSSBSS is shown in Figure 4-561 and described in Table 4-590.

Return to the Table 4-541.

Figure 4-561. ECCCAPTMSSBSS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTMSSBSS																															
R-0h																															

Table 4-590. ECCCAPTMSSBSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTMSSBSS	R	0h	7:0 : mss_mbox4bss_ecc_fault_address 14:8 : mss_mbox4bss_repaired_bit 23:16 : bss_mbox4mss_ecc_fault_address 30:24 : bss_mbox4mss_repaired_bit

4.8.2.50 CLKDIVCTL2 Register (Offset = F0h) [reset = 0h]

CLKDIVCTL2 is shown in Figure 4-562 and described in Table 4-591.

Return to the Table 4-541.

Figure 4-562. CLKDIVCTL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								QSPICLKDIV							
0h								R/W-0h							

Table 4-591. CLKDIVCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	
7-0	QSPICLKDIV	R/W	0h	Divide value for QSPI baud clock selected by field QSPICLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256

4.8.2.51 QSPIMCONNECT Register (Offset = F4h) [reset = 103h]

QSPIMCONNECT is shown in Figure 4-563 and described in Table 4-592.

Return to the Table 4-541.

Figure 4-563. QSPIMCONNECT Register

31	30	29	28	27	26	25	24
NU2							
0h							
23	22	21	20	19	18	17	16
NU2							
0h							
15	14	13	12	11	10	9	8
NU2							FCLKEN
0h							R/W-1h
7	6	5	4	3	2	1	0
NU1						MCONNECT	
0h						R/W-3h	

Table 4-592. QSPIMCONNECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	NU2		0h	
8	FCLKEN	R/W	1h	Not Used.Reserved
7-2	NU1		0h	
1-0	MCONNECT	R/W	3h	Not Used.Reserved

4.8.2.52 QSPISCONNECT Register (Offset = F8h) [reset = 0h]

QSPISCONNECT is shown in Figure 4-564 and described in Table 4-593.

Return to the Table 4-541.

Figure 4-564. QSPISCONNECT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3													SCONNECT		
0h													R-0h		

Table 4-593. QSPISCONNECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU3		0h	
2-0	SCONNECT	R	0h	Not Used.Reserved

4.8.2.53 SWIRQC Register (Offset = FCh) [reset = 0h]

SWIRQC is shown in Figure 4-565 and described in Table 4-594.

Return to the Table 4-541.

Figure 4-565. SWIRQC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ5								SWIRQ5DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ4								SWIRQ4DAT							
R/W-0h								R/W-0h							

Table 4-594. SWIRQC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ5	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ5DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ4	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ4DAT	R/W	0h	Not Used.Reserved

4.8.3 MSS_GPCFG_REG Registers

Table 4-891 lists the memory-mapped registers for the MSS_GPCFG_REG. All register offset addresses not listed in Table 4-891 should be considered as reserved locations and the register contents should not be modified.

Table 4-595. MSS_GPCFG_REG Registers

Offset	Acronym	Register Name	Section
0h	GPCFG0	GPCFG0	Section 4.9.3.1
4h	GPCFG1	GPCFG1	Section 4.9.3.2
8h	GPCFG2	GPCFG2	Section 4.9.3.3
Ch	GPCFG3	GPCFG3	Section 4.9.3.4
10h	GPCFG4	GPCFG4	Section 4.9.3.5
2Ch	GPCFG11	GPCFG11	Section 4.9.3.7
D0h	CCCACFG0	CCCA_CFG0	Section 4.9.3.8
D4h	CCCACFG1	CCCA_CFG1	Section 4.9.3.9
D8h	CCCACFG2	CCCA_CFG2	Section 4.9.3.10
DCh	CCCACFG3	CCCA_CFG3	Section 4.9.3.11
E0h	CCCBCFG0	CCCB_CFG0	Section 4.9.3.12
E4h	CCCBCFG1	CCCB_CFG1	Section 4.9.3.13
E8h	CCCBCFG2	CCCB_CFG2	Section 4.9.3.14
ECh	CCCBCFG3	CCCB_CFG3	Section 4.9.3.15
F0h	CCCACNTVAL	CCCACNTVAL	Section 4.9.3.16
F4h	CCCBCNTVAL	CCCBCNTVAL	Section 4.9.3.17
F8h	CCCABERRSTAT	CCCABERRSTAT	Section 4.9.3.18
FCh	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.3.19
140h	EPWMCFG	EPWMCFG	Section 4.9.3.20
148h	DMMSWINT0	DMMSWINT0	Section 4.9.3.21
14Ch	DMMSWINT1	DMMSWINT1	Section 4.9.3.22
150h	DMMSWINTSEL0	DMMSWINTSEL0	Section 4.9.3.23
154h	DMMSWINTSEL1	DMMSWINTSEL1	Section 4.9.3.24
158h	CCCBWDEN	CCCBWDEN	Section 4.9.3.25
15Ch	GPIOINTREDGESEL	GPIOINTREDGESEL	Section 4.9.3.26
164h	PWMDMATRIGEN	PWMDMATRIGEN	Section 4.9.3.27
168h	JTAGTXDATA	JTAGTXDATA	Section 4.9.3.28
16Ch	JTAGTXCONTROL	JTAGTXCONTROL	Section 4.9.3.29
170h	JTAGRXDATA	JTAGRXDATA	Section 4.9.3.30
178h	JTAGTXRXACK	JTAGTXRXACK	Section 4.9.3.31
17Ch	JTAGRXCONTROL	JTAGRXCONTROL	Section 4.9.3.32
180h	MSS2GEMSWIRQ		Section 4.9.3.33
184h	CSETBFLUSH		Section 4.9.3.34

Complex bit access types are encoded to fit into small table cells. Table 4-892 shows the codes that are used for access types in this section.

Table 4-596. MSS_GPCFG_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write

Table 4-596. MSS_GPCFG_REG Access Type Codes (continued)

Access Type	Code	Description
Reset or Default Value		
-n		Value after reset or the default value

4.8.3.1 GPCFG0 Register (Offset = 0h) [reset = 0h]

GPCFG0 is shown in Figure 4-852 and described in Table 4-893.

Return to Summary Table.

Figure 4-566. GPCFG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG0																															
R/W-0h																															

Table 4-597. GPCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG0	R/W	0h	General Purpose config register for SW use.

4.8.3.2 GPCFG1 Register (Offset = 4h) [reset = 0h]

GPCFG1 is shown in Figure 4-853 and described in Table 4-894.

Return to Summary Table.

Figure 4-567. GPCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG1																															
R/W-0h																															

Table 4-598. GPCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG1	R/W	0h	General Purpose config register for SW use.

4.8.3.3 GPCFG2 Register (Offset = 8h) [reset = 0h]

GPCFG2 is shown in Figure 4-854 and described in Table 4-895.

Return to Summary Table.

Figure 4-568. GPCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG2																															
R/W-0h																															

Table 4-599. GPCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG2	R/W	0h	General Purpose config register for SW use.

4.8.3.4 GPCFG3 Register (Offset = Ch) [reset = 0h]

GPCFG3 is shown in Figure 4-855 and described in Table 4-896.

Return to Summary Table.

Figure 4-569. GPCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG3																															
R/W-0h																															

Table 4-600. GPCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG3	R/W	0h	General Purpose config register for SW use.

4.8.3.5 GPCFG4 Register (Offset = 10h) [reset = 0h]

GPCFG4 is shown in Figure 4-856 and described in Table 4-897.

Return to Summary Table.

Figure 4-570. GPCFG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG4																															
R/W-0h																															

Table 4-601. GPCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG4	R/W	0h	General Purpose config register for SW use.

4.8.3.6 GPCFG6 Register (Offset = 18h) [reset = 0h]

GPCFG6 is shown in Figure 4-857 and described in Table 4-898.

Return to Summary Table.

Figure 4-571. GPCFG6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG6																															
R/W-0h																															

Table 4-602. GPCFG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG6	R/W	0h	Describes the muxing of Interrupts and DMA based on use cases - GPCFG6[0] = 1 : dma_req[36] = can_fd_intr[0] ,0 : GPCFG6[0] = 0 : dma_req[36] = can_fd_fe_intr[0] GPCFG6[1] = 1 : dma_req[44] = can_fd_intr[1] ,0 : GPCFG6[1] = 0 : dma_req[44] = can_fd_fe_intr[1] GPCFG6[2] = 1 : frc_can_intr[3] = can_fd_fe_intr[0] ,0 : GPCFG6[2] = 0 : frc_can_intr[3] = can_fd_intr[0] GPCFG6[3] = 1 : frc_can_intr[2] = can_fd_fe_intr[1] ,0 : GPCFG6[3] = 0 : frc_can_intr[2] = can_fd_intr[1] GPCFG6[8] = 1 : dma_req[5] = can_fd_fe_intr[2] ,0 : GPCFG6[8] = 0 : dma_req[5] = spia_dma_req[3] GPCFG6[11] = 1 : dma_req[37] = rti2 dma req[0] ,0 : GPCFG6[11] = 0 : dma_req[37] = spib_dma_req[2] GPCFG6[12] = 1 : dma_req[38] = rti2 dma req[1] ,0 : GPCFG6[12] = 0 : dma_req[38] = else spib_dma_req[3] GPCFG6[13] = 1 : dma_req[42] = rti3 dma req[0] ,0 : GPCFG6[13] = 0 : dma_req[42] = else spib_dma_req[4] GPCFG6[14] = 1 : dma_req[43] = rti3 dma req[1] ,0 : GPCFG6[14] = 0 : dma_req[43] = else spib_dma_req[5] GPCFG6[26] = 1 : irq_req[108] = rti2_int_req[0] ,0 : GPCFG6[26] = 0 : irq_req[108] = else epwm2_int1 GPCFG6[27] = 1 : irq_req[109] = rti2_int_req[1] ,0 : GPCFG6[27] = 0 : irq_req[109] = else epwm2_int2 GPCFG6[28] = 1 : irq_req[110] = rti3_int_req[0] ,0 : GPCFG6[28] = 0 : irq_req[110] = else epwm3_int1 GPCFG6[29] = 1 : irq_req[111] = rti3_int_req[1] ,0 : GPCFG6[29] = 0 : irq_req[111] = else epwm3_int2

4.8.3.7 GPCFG11 Register (Offset = 2Ch) [reset = 0h]

GPCFG11 is shown in Figure 4-858 and described in Table 4-899.

Return to Summary Table.

Figure 4-572. GPCFG11 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED						BSS2DSSSWIRQ2	BSS2DSSSWIRQ1
R-0h						0h	0h
15	14	13	12	11	10	9	8
RESERVED						DSS2BSSSWIRQ2	DSS2BSSSWIRQ1
R-0h						0h	0h
7	6	5	4	3	2	1	0
RESERVED						MSS2BSSSWIRQ2	MSS2BSSSWIRQ1
R-0h						0h	0h

Table 4-603. GPCFG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED		0h	Reserved
17	BSS2DSSSWIRQ2		0h	Self clearing register bit to generate interrupt to DSP from BSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
16	BSS2DSSSWIRQ1		0h	Self clearing register bit to generate interrupt to DSP from BSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
15-10	RESERVED	R	0h	Reserved
9	DSS2BSSSWIRQ2		0h	Self clearing register bit to generate interrupt to BSS from DSP. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
8	DSS2BSSSWIRQ1		0h	Self clearing register bit to generate interrupt to BSS from DSP. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
7-2	RESERVED	R	0h	Reserved
1	MSS2BSSSWIRQ2		0h	Self clearing register bit to generate interrupt to BSS from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
0	MSS2BSSSWIRQ1		0h	Self clearing register bit to generate interrupt to BSS from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS

4.8.3.8 CCCACFG0 Register (Offset = D0h) [reset = 0h]

CCCACFG0 is shown in Figure 4-859 and described in Table 4-900.

Return to Summary Table.

Figure 4-573. CCCACFG0 Register

31	30	29	28	27	26	25	24
CCCA_MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
CCCA_MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU31							CCCA_SINGLE_SHOT_MODE
0h							R/W-0h
7	6	5	4	3	2	1	0
CCCA_ENABLE_MODULE	CCCA_DISABLE_CLOCKS	CCCA_CLOCK1_SEL			CCCA_CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-604. CCCACFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CCCA_MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU31		0h	Reserved
8	CCCA_SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	CCCA_ENABLE_MODULE	R/W	0h	Enable for CCC
6	CCCA_DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CCCA_CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CCCA_CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.8.3.9 CCCACFG1 Register (Offset = D4h) [reset = 0h]

CCCACFG1 is shown in Figure 4-860 and described in Table 4-901.

Return to Summary Table.

Figure 4-574. CCCACFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG1																															
R/W-0h																															

Table 4-605. CCCACFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG1	R/W	0h	count0_expiry_val

4.8.3.10 CCCACFG2 Register (Offset = D8h) [reset = 0h]

CCCACFG2 is shown in Figure 4-861 and described in Table 4-902.

Return to Summary Table.

Figure 4-575. CCCACFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG2																															
R/W-0h																															

Table 4-606. CCCACFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG2	R/W	0h	count1_expected_val

4.8.3.11 CCCACFG3 Register (Offset = DCh) [reset = 0h]

CCCACFG3 is shown in Figure 4-862 and described in Table 4-903.

Return to Summary Table.

Figure 4-576. CCCACFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG3																															
R/W-0h																															

Table 4-607. CCCACFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG3	R/W	0h	Error Counter value in counter1 clock

4.8.3.12 CCCBCFG0 Register (Offset = E0h) [reset = 0h]

CCCBCFG0 is shown in Figure 4-863 and described in Table 4-904.

Return to Summary Table.

Figure 4-577. CCCBCFG0 Register

31	30	29	28	27	26	25	24
CCCB_MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
CCCB_MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU32							CCCB_SINGLE_SHOT_MODE
0h							R/W-0h
7	6	5	4	3	2	1	0
CCCB_ENABLE_MODULE	CCCB_DISABLE_CLOCKS	CCCB_CLOCK1_SEL			CCCB_CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-608. CCCBCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CCCB_MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU32		0h	Reserved
8	CCCB_SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	CCCB_ENABLE_MODULE	R/W	0h	Enable for CCC
6	CCCB_DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CCCB_CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CCCB_CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.8.3.13 CCCBCFG1 Register (Offset = E4h) [reset = 0h]

CCCBCFG1 is shown in Figure 4-864 and described in Table 4-905.

Return to Summary Table.

Figure 4-578. CCCBCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG1																															
R/W-0h																															

Table 4-609. CCCBCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG1	R/W	0h	count0_expiry_val

4.8.3.14 CCCBCFG2 Register (Offset = E8h) [reset = 0h]

CCCBCFG2 is shown in Figure 4-865 and described in Table 4-906.

Return to Summary Table.

Figure 4-579. CCCBCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG2																															
R/W-0h																															

Table 4-610. CCCBCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG2	R/W	0h	count1_expected_val

4.8.3.15 CCCBCFG3 Register (Offset = ECh) [reset = 0h]

CCCBCFG3 is shown in Figure 4-866 and described in Table 4-907.

Return to Summary Table.

Figure 4-580. CCCBCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG3																															
R/W-0h																															

Table 4-611. CCCBCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG3	R/W	0h	Error Counter value in counter1 clock

4.8.3.16 CCCACNTVAL Register (Offset = F0h) [reset = 0h]

CCCACNTVAL is shown in Figure 4-867 and described in Table 4-908.

Return to Summary Table.

Figure 4-581. CCCACNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACNTVAL																															
R-0h																															

Table 4-612. CCCACNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACNTVAL	R	0h	count1_val_out

4.8.3.17 CCCBCNTVAL Register (Offset = F4h) [reset = 0h]

CCCBCNTVAL is shown in Figure 4-868 and described in Table 4-909.

Return to Summary Table.

Figure 4-582. CCCBCNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCNTVAL																															
R-0h																															

Table 4-613. CCCBCNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCNTVAL	R	0h	count1_val_out

4.8.3.18 CCCABERRSTAT Register (Offset = F8h) [reset = 0h]

CCCABERRSTAT is shown in Figure 4-869 and described in Table 4-910.

Return to Summary Table.

Figure 4-583. CCCABERRSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCABERRSTAT																															
R-0h																															

Table 4-614. CCCABERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCABERRSTAT	R	0h	7:0 : CCCA Error Status 15:8 : CCCB Error Status

4.8.3.19 USERMODEEN Register (Offset = FCh) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-584. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-615. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS GPCFG space.

4.8.3.20 EPWMCFG Register (Offset = 140h) [reset = F00h]

EPWMCFG is shown in Figure 4-871 and described in Table 4-912.

Return to Summary Table.

Figure 4-585. EPWMCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPWMCFG																															
R/W-F00h																															

Table 4-616. EPWMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EPWMCFG	R/W	F00h	[1:0] : 0->rampgen Frame start is SYNCIN to EPWM1 1->FRC Frame Start is SYNCIN to EPWM1 2,3-> External SYNCIN is SYNCIN to EPWM1 [3:2]: 0->rampgen Frame start is SYNCIN to EPWM2 1->FRC Frame Start is SYNCIN to EPWM2 2,3-> EPWM1 SYNCO is SYNCIN to EPWM2 [5:4]: 0->rampgen Frame start is SYNCIN to EPWM3 1->FRC Frame Start is SYNCIN to EPWM3 2,3-> EPWM2 SYNCO is SYNCIN to EPWM3 [31:8] : Reserved

4.8.3.21 DMMSWINT0 Register (Offset = 148h) [reset = 0h]

DMMSWINT0 is shown in Figure 4-872 and described in Table 4-913.

Return to Summary Table.

Figure 4-586. DMMSWINT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINT0																															
0h																															

Table 4-617. DMMSWINT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINT0		0h	HIL Intr0 muxed with frame start interrupt to DSP HIL Intr1 muxed with logical frame start interrupt to DSP HIL Intr2 muxed with Ping/Pong threshold trigger interrupt for DSP HIL Intr3 muxed with ADC data valid interrupt for DSP HIL Intr4 - HIL Intr8 : SW interrupt for DSP HIL Intr9 muxed with frame start interrupt to TPCC0 HIL Intr10 muxed with logical frame start interrupt to TPCC0 HIL Intr11 muxed with Ping/Pong threshold trigger interrupt for TPCC0 HIL Intr12 muxed with ADC data valid interrupt for TPCC0 HIL Intr13 - HIL Intr18 as SW interrupt for TPCC0 HIL Intr19 muxed with frame start interrupt to TPCC1 HIL Intr20 muxed with logical frame start interrupt to TPCC1 HIL Intr21 muxed with Ping/Pong threshold trigger interrupt for TPCC1 HIL Intr22 muxed with ADC data valid interrupt for TPCC1 HIL Intr23 - HIL Intr28 as SW interrupt for TPCC1 HIL Intr29 muxed with frame start interrupt to VIMMR4F HIL Intr30 muxed with logical frame start interrupt to VIMMR4F HIL Intr31 muxed with Ping/Pong threshold trigger interrupt for VIMMR4F HIL Intr32 muxed with ADC data valid interrupt for VIMMR4F HIL Intr33 - HIL Intr38 as SW interrupt for VIMMR4F HIL Intr39 muxed with frame start interrupt right at the source - propagates throughout the system as actual FRC output does. This is muxed with the frame start coming from BSS to TOP so that it propagates to MSS/DSS. HIL Intr40 muxed with frame start interrupt right at the source - propagates throughout the system as actual FRC output does. This is muxed with the ADC clock enable signal coming from FRC. Intr41 muxed with logical frame start interrupt right at the source (FRC) - propagates throughout the system Intr42 muxed with logical frame end interrupt right at the source (FRC) - propagates throughout the system Intr43 muxed with Ping/Pong threshold trigger interrupt right at the source - propagates throughout the system Intr44 muxed with ADC data valid interrupt right at the source - propagates throughout the system

4.8.3.22 DMMSWINT1 Register (Offset = 14Ch) [reset = 0h]

DMMSWINT1 is shown in Figure 4-953 and described in Table 4-996.

Return to Summary Table.

Figure 4-587. DMMSWINT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINT1																															
0h																															

Table 4-618. DMMSWINT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINT1		0h	HIL Interrupts - MSB 32 bits [63:32]

4.8.3.23 DMMSWINTSEL0 Register (Offset = 150h) [reset = 0h]

DMMSWINTSEL0 is shown in Figure 4-874 and described in Table 4-915.

Return to Summary Table.

Figure 4-588. DMMSWINTSEL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINTSEL0																															
R/W-0h																															

Table 4-619. DMMSWINTSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINTSEL0	R/W	0h	Mux control for HIL Interrupts - LSB 32 bits

4.8.3.24 DMMSWINTSEL1 Register (Offset = 154h) [reset = 0h]

DMMSWINTSEL1 is shown in Figure 4-875 and described in Table 4-916.

Return to Summary Table.

Figure 4-589. DMMSWINTSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINTSEL1																															
R/W-0h																															

Table 4-620. DMMSWINTSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINTSEL1	R/W	0h	Mux control for HIL Interrupts - MSB 32 bits

4.8.3.25 CCCBWDEN Register (Offset = 158h) [reset = 0h]

CCCBWDEN is shown in Figure 4-876 and described in Table 4-917.

Return to Summary Table.

Figure 4-590. CCCBWDEN Register

31	30	29	28	27	26	25	24
NU36							
R-0h							
23	22	21	20	19	18	17	16
NU36							ENABLECCBERRRSTN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU35							
R-0h							
7	6	5	4	3	2	1	0
NU35							ENABLECCBERRNMI
R-0h							R/W-0h

Table 4-621. CCCBWDEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU36	R	0h	Reserved
16	ENABLECCBERRRSTN	R/W	0h	Enable CCCB error to generate WD restn.In this mode CCCB has to compare MSS CR4 clock to XTAL
15-1	NU35	R	0h	Reserved
0	ENABLECCBERRNMI	R/W	0h	Enable CCCB error to generate NMI.In this mode CCCB has to compare MSS CR4 clock to XTAL

4.8.3.26 GPIOINTREDGESEL Register (Offset = 15Ch) [reset = 0h]

GPIOINTREDGESEL is shown in Figure 4-877 and described in Table 4-918.

Return to Summary Table.

Figure 4-591. GPIOINTREDGESEL Register

31	30	29	28	27	26	25	24
NU37							
0h							
23	22	21	20	19	18	17	16
NU37							
0h							
15	14	13	12	11	10	9	8
NU37							
0h							
7	6	5	4	3	2	1	0
NU37					GPIO2EDGESEL	GPIO1EDGESEL	GPIO0EDGESEL
0h					R/W-0h	R/W-0h	R/W-0h

Table 4-622. GPIOINTREDGESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU37		0h	Reserved
2	GPIO2EDGESEL	R/W	0h	0:Posedge of GPIO2 generates interruptt to MSS CR4 and DSP 1:Negedge
1	GPIO1EDGESEL	R/W	0h	0:Posedge of GPIO1 generates interruptt to MSS CR4 and DSP 1:Negedge
0	GPIO0EDGESEL	R/W	0h	0:Posedge of GPIO0 generates interruptt to MSS CR4 and DSP 1:Negedge

4.8.3.27 PWMDMATRIGEN Register (Offset = 164h) [reset = 0h]

PWMDMATRIGEN is shown in Figure 4-878 and described in Table 4-919.

Return to Summary Table.

Figure 4-592. PWMDMATRIGEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU38															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU38												PWMDMATRIGEN			
R-0h												R/W-0h			

Table 4-623. PWMDMATRIGEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU38	R	0h	Reserved
3-0	PWMDMATRIGEN	R/W	0h	0: "1" -> Mux epwm1a instead of mss_event_gen_1_frc to DMA 1: "1" -> Mux epwm1b instead of mss_event_gen_1_frc to DMA 2: "1" -> Mux epwm2a instead of mss_event_gen_1_frc to DMA 3: "1" -> Mux epwm2b instead of mss_event_gen_1_frc to DMA

4.8.3.28 JTAGTXDATA Register (Offset = 168h) [reset = 0h]

JTAGTXDATA is shown in Figure 4-879 and described in Table 4-920.

Return to Summary Table.

Figure 4-593. JTAGTXDATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGTXDATA																															
R-0h																															

Table 4-624. JTAGTXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGTXDATA	R	0h	Transmit Data. This register is used to pass data to the system security logic

4.8.3.29 JTAGTXCONTROL Register (Offset = 16Ch) [reset = 0h]

JTAGTXCONTROL is shown in Figure 4-880 and described in Table 4-921.

Return to Summary Table.

Figure 4-594. JTAGTXCONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGTXCONTROL																															
R-0h																															

Table 4-625. JTAGTXCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGTXCONTROL	R	0h	Provides the handshake for the JTAGTXDATA Register and can also be used to pass control information to the system security logic. Only bits [31:1] are valid

4.8.3.30 JTAGRXDATA Register (Offset = 170h) [reset = 0h]

JTAGRXDATA is shown in Figure 4-881 and described in Table 4-922.

Return to Summary Table.

Figure 4-595. JTAGRXDATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGRXDATA																															
R/W-0h																															

Table 4-626. JTAGRXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGRXDATA	R/W	0h	Receive data. This register is used to pass data from the system security logic

4.8.3.31 JTAGTXRXACK Register (Offset = 178h) [reset = 0h]

JTAGTXRXACK is shown in Figure 4-882 and described in Table 4-923.

Return to Summary Table.

Figure 4-596. JTAGTXRXACK Register

31	30	29	28	27	26	25	24
NU40							
R-0h							
23	22	21	20	19	18	17	16
NU40							
R-0h							
15	14	13	12	11	10	9	8
NU40							JTAGRXDATAWR
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NU39							JTAGTXDATARD
R-0h							R/W-0h

Table 4-627. JTAGTXRXACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	NU40	R	0h	Reserved
8	JTAGRXDATAWR	R/W	0h	Indication from system security logic that JTAGRXDATA is valid
7-1	NU39	R	0h	Reserved
0	JTAGTXDATARD	R/W	0h	Indication from the system security logic that JTAGTXDATA has been accepted

4.8.3.32 JTAGRXCONTROL Register (Offset = 17Ch) [reset = 0h]

JTAGRXCONTROL is shown in Figure 4-883 and described in Table 4-924.

Return to Summary Table.

Figure 4-597. JTAGRXCONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGRXCONTROL																															
R/W-0h																															

Table 4-628. JTAGRXCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGRXCONTROL	R/W	0h	Provides the handshake for the JTAGRXDATA Register and can also be used to pass control information from the system security logic. Only bits [31:1] is used.

4.8.3.33 MSS2GEMSWIRQ Register (Offset = 180h) [reset = 0h]

MSS2GEMSWIRQ is shown in Figure 4-884 and described in Table 4-925.

Return to Summary Table.

Figure 4-598. MSS2GEMSWIRQ Register

31	30	29	28	27	26	25	24
NU41							
R-0h							
23	22	21	20	19	18	17	16
NU41							
R-0h							
15	14	13	12	11	10	9	8
NU41							
R-0h							
7	6	5	4	3	2	1	0
NU41						MSS2GEMSWIRQ2	MSS2GEMSWIRQ1
R-0h						0h	0h

Table 4-629. MSS2GEMSWIRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU41	R	0h	Reserved
1	MSS2GEMSWIRQ2		0h	Self clearing register bit to generate interrupt to DSP from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
0	MSS2GEMSWIRQ1		0h	Self clearing register bit to generate interrupt to DSP from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP

4.8.3.34 CSETBFLUSH Register (Offset = 184h) [reset = 0h]

CSETBFLUSH is shown in Figure 4-885 and described in Table 4-926.

Return to Summary Table.

Figure 4-599. CSETBFLUSH Register

31	30	29	28	27	26	25	24
NU43							
R-0h							
23	22	21	20	19	18	17	16
NU43							
R-0h							
15	14	13	12	11	10	9	8
NU43					CSETBFULL	CSETBACQ_OMPLETE	CSETBFLUSHINACK
R-0h					R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
NU42							CSETBFLUSHIN
R-0h							R/W-0h

Table 4-630. CSETBFLUSH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU43	R	0h	Reserved
10	CSETBFULL	R	0h	When HIGH indicates that the ETB RAM has overflowed or wrapped around to address zero
9	CSETBACQ_OMPLETE	R	0h	When HIGH, indicates that trace acquisition is complete by ETB, that is, the trigger counter is at zero
8	CSETBFLUSHINACK	R	0h	Return acknowledgement to CSETBFLUSHIN
7-1	NU42	R	0h	Reserved
0	CSETBFLUSHIN	R/W	0h	External control used to assert the ATB signal AFVALIDS and drain any historical FIFO information on the bus

4.8.4 DSS_REG Registers

Table 4-927 lists the memory-mapped registers for the DSS_REG. All register offset addresses not listed in Table 4-927 should be considered as reserved locations and the register contents should not be modified.

Table 4-631. DSS_REG Registers

Offset	Acronym	Register Name	Section
50h	RTIEVENTCAPTURESEL	RTIEVENTCAPTURESEL	Section 4.9.4.1
6Ch	CQCFG1	CQCFG1	Section 4.9.4.2
80h	TPCCPARSTATCFG	TPCCPARSTATCFG	Section 4.9.4.3
104h	TPTC0WRMPUSTADD0	TPTC0WRMPUSTADD0	Section 4.9.4.4
108h	TPTC0WRMPUSTADD1	TPTC0WRMPUSTADD1	Section 4.9.4.5
10Ch	TPTC0WRMPUSTADD2	TPTC0WRMPUSTADD2	Section 4.9.4.6
110h	TPTC0WRMPUSTADD3	TPTC0WRMPUSTADD3	Section 4.9.4.7
114h	TPTC0WRMPUSTADD4	TPTC0WRMPUSTADD4	Section 4.9.4.8
118h	TPTC0WRMPUSTADD5	TPTC0WRMPUSTADD5	Section 4.9.4.9
124h	TPTC0WRMPUENDADD0	TPTC0WRMPUENDADD0	Section 4.9.4.10
128h	TPTC0WRMPUENDADD1	TPTC0WRMPUENDADD1	Section 4.9.4.11
12Ch	TPTC0WRMPUENDADD2	TPTC0WRMPUENDADD2	Section 4.9.4.12
130h	TPTC0WRMPUENDADD3	TPTC0WRMPUENDADD3	Section 4.9.4.13
134h	TPTC0WRMPUENDADD4	TPTC0WRMPUENDADD4	Section 4.9.4.14
138h	TPTC0WRMPUENDADD5	TPTC0WRMPUENDADD5	Section 4.9.4.15
144h	TPTC0WRMPUERRADD	TPTC0WRMPUERRADD	Section 4.9.4.16
148h	TPTC0RDMPUSTADD0	TPTC0RDMPUSTADD0	Section 4.9.4.17
14Ch	TPTC0RDMPUSTADD1	TPTC0RDMPUSTADD1	Section 4.9.4.18
150h	TPTC0RDMPUSTADD2	TPTC0RDMPUSTADD2	Section 4.9.4.19
154h	TPTC0RDMPUSTADD3	TPTC0RDMPUSTADD3	Section 4.9.4.20
158h	TPTC0RDMPUSTADD4	TPTC0RDMPUSTADD4	Section 4.9.4.21
15Ch	TPTC0RDMPUSTADD5	TPTC0RDMPUSTADD5	Section 4.9.4.22
168h	TPTC0RDMPUENDADD0	TPTC0RDMPUENDADD0	Section 4.9.4.23
16Ch	TPTC0RDMPUENDADD1	TPTC0RDMPUENDADD1	Section 4.9.4.24
170h	TPTC0RDMPUENDADD2	TPTC0RDMPUENDADD2	Section 4.9.4.25
174h	TPTC0RDMPUENDADD3	TPTC0RDMPUENDADD3	Section 4.9.4.26
178h	TPTC0RDMPUENDADD4	TPTC0RDMPUENDADD4	Section 4.9.4.27
17Ch	TPTC0RDMPUENDADD5	TPTC0RDMPUENDADD5	Section 4.9.4.28
188h	TPTC0RDMPUERRADD	TPTC0RDMPUERRADD	Section 4.9.4.29
18Ch	TPTC1WRMPUSTADD0	TPTC1WRMPUSTADD0	Section 4.9.4.30
190h	TPTC1WRMPUSTADD1	TPTC1WRMPUSTADD1	Section 4.9.4.31
194h	TPTC1WRMPUSTADD2	TPTC1WRMPUSTADD2	Section 4.9.4.32
198h	TPTC1WRMPUSTADD3	TPTC1WRMPUSTADD3	Section 4.9.4.33
19Ch	TPTC1WRMPUSTADD4	TPTC1WRMPUSTADD4	Section 4.9.4.34
1A0h	TPTC1WRMPUSTADD5	TPTC1WRMPUSTADD5	Section 4.9.4.35
1ACh	TPTC1WRMPUENDADD0	TPTC1WRMPUENDADD0	Section 4.9.4.36
1B0h	TPTC1WRMPUENDADD1	TPTC1WRMPUENDADD1	Section 4.9.4.37
1B4h	TPTC1WRMPUENDADD2	TPTC1WRMPUENDADD2	Section 4.9.4.38
1B8h	TPTC1WRMPUENDADD3	TPTC1WRMPUENDADD3	Section 4.9.4.39
1BCh	TPTC1WRMPUENDADD4	TPTC1WRMPUENDADD4	Section 4.9.4.40
1C0h	TPTC1WRMPUENDADD5	TPTC1WRMPUENDADD5	Section 4.9.4.41
1CCh	TPTC1WRMPUERRADD	TPTC1WRMPUERRADD	Section 4.9.4.42
1D0h	TPTC1RDMPUSTADD0	TPTC1RDMPUSTADD0	Section 4.9.4.43

Table 4-631. DSS_REG Registers (continued)

Offset	Acronym	Register Name	Section
1D4h	TPTC1RDMPUSTADD1	TPTC1RDMPUSTADD1	Section 4.9.4.44
1D8h	TPTC1RDMPUSTADD2	TPTC1RDMPUSTADD2	Section 4.9.4.45
1DCh	TPTC1RDMPUSTADD3	TPTC1RDMPUSTADD3	Section 4.9.4.46
1E0h	TPTC1RDMPUSTADD4	TPTC1RDMPUSTADD4	Section 4.9.4.47
1E4h	TPTC1RDMPUSTADD5	TPTC1RDMPUSTADD5	Section 4.9.4.48
1F0h	TPTC1RDMPUENDADD0	TPTC1RDMPUENDADD0	Section 4.9.4.49
1F4h	TPTC1RDMPUENDADD1	TPTC1RDMPUENDADD1	Section 4.9.4.50
1F8h	TPTC1RDMPUENDADD2	TPTC1RDMPUENDADD2	Section 4.9.4.51
1FCh	TPTC1RDMPUENDADD3	TPTC1RDMPUENDADD3	Section 4.9.4.52
200h	TPTC1RDMPUENDADD4	TPTC1RDMPUENDADD4	Section 4.9.4.53
204h	TPTC1RDMPUENDADD5	TPTC1RDMPUENDADD5	Section 4.9.4.54
210h	TPTC1RDMPUERRADD	TPTC1RDMPUERRADD	Section 4.9.4.55
214h	TPTCMPUVALIDCFG	TPTCMPUVALIDCFG	Section 4.9.4.56
218h	TPTCMPUENCFG	TPTCMPUENCFG	Section 4.9.4.57
21Ch	TESTPATTERNRX1ICFG	TESTPATTERNRX1ICFG	Section 4.9.4.58
220h	TESTPATTERNRX2ICFG	TESTPATTERNRX2ICFG	Section 4.9.4.59
224h	TESTPATTERNRX3ICFG	TESTPATTERNRX3ICFG	Section 4.9.4.60
228h	TESTPATTERNRX4ICFG	TESTPATTERNRX4ICFG	Section 4.9.4.61
22Ch	TESTPATTERNRX1QCFG	TESTPATTERNRX1QCFG	Section 4.9.4.62
230h	TESTPATTERNRX2QCFG	TESTPATTERNRX2QCFG	Section 4.9.4.63
234h	TESTPATTERNRX3QCFG	TESTPATTERNRX3QCFG	Section 4.9.4.64
238h	TESTPATTERNRX4QCFG	TESTPATTERNRX4QCFG	Section 4.9.4.65
23Ch	TESTPATTERNVLDCFG	TESTPATTERNVLDCFG	Section 4.9.4.66
240h	DSSMISC	DSSMISC	
258h	TPCC1PARSTATCFG	TPCC1PARSTATCFG	Section 4.9.4.67
260h	DMMSWINT1	DMMSWINT1	Section 4.9.4.68
270h	DSSINTRCFG	DSSINTRCFG	Section 4.9.4.69
274h	MPUMSTIDCFG1	MPUMSTIDCFG1	Section 4.9.4.70
278h	MPUMSTIDCFG2	MPUMSTIDCFG2	Section 4.9.4.71
27Ch	MPUMSTIDCFG3	MPUMSTIDCFG3	Section 4.9.4.72
280h	HSRAM1ECCCFG	HSRAM1ECCCFG	Section 4.9.4.73
288h	DATATRRAMECCCFG	DATATRRAMECCCFG	Section 4.9.4.74
28Ch	ADCBUFPINGECCCFG	ADCBUFPINGECCCFG	Section 4.9.4.75
290h	ADCBUFPONGECCCFG	ADCBUFPONGECCCFG	Section 4.9.4.76
29Ch	UMAP0PARITYCFG1	UMAP0PARITYCFG1	Section 4.9.4.77
2A0h	UMAP0PARITYCFG2	UMAP0PARITYCFG2	Section 4.9.4.78
2A4h	UMAP0PARITYCFG3	UMAP0PARITYCFG3	Section 4.9.4.79
2A8h	UMAP1PARITYCFG1	UMAP1PARITYCFG1	Section 4.9.4.80
2ACh	UMAP1PARITYCFG2	UMAP1PARITYCFG2	Section 4.9.4.81
2B0h	UMAP1PARITYCFG3	UMAP1PARITYCFG3	Section 4.9.4.82
2B4h	ESMGRP2MASKCFG	ESMGRP2MASKCFG	Section 4.9.4.83
2B8h	L2MEMINITCFG1[(1)]	L2MEMINITCFG1	Section 4.9.4.84
2BCh	L2MEMINITCFG2[(1)]	L2MEMINITCFG2	Section 4.9.4.85
2C0h	GEMRSTCAUSE[(1)]	GEMRSTCAUSE	Section 4.9.4.86
2CCh	GEMPWRSMCFG4[(1)]	GEMPWRSMCFG4	Section 4.9.4.87
2D4h	PWRSMWAKEMASK0[(1)]	PWRSMWAKEMASK0	Section 4.9.4.88

(1) Applicable for devices with DSP C674x (such as the 68xx device).

Table 4-631. DSS_REG Registers (continued)

Offset	Acronym	Register Name	Section
2D8h	PWRSMWAKEMASK1[(1)]	PWRSMWAKEMASK1	Section 4.9.4.89
2DCh	PWRSMWAKEMASK2[(1)]	PWRSMWAKEMASK2	Section 4.9.4.90
2E0h	PWRSMMISEVTMASK0[(1)]	PWRSMMISEVTMASK0	Section 4.9.4.91
2E4h	PWRSMMISEVTMASK1[(1)]	PWRSMMISEVTMASK1	Section 4.9.4.92
2E8h	PWRSMMISEVTMASK2[(1)]	PWRSMMISEVTMASK2	Section 4.9.4.93
2ECh	PWRSMWAKESRCSTAT0[(1)]	PWRSMWAKESRCSTAT0	Section 4.9.4.94
2F0h	PWRSMWAKESRCSTAT1[(1)]	PWRSMWAKESRCSTAT1	Section 4.9.4.95
320h	PWRSMWAKESRCSTAT2[(1)]	PWRSMWAKESRCSTAT2	Section 4.9.4.96
324h	PWRSMEVNTMONSTAT0[(1)]	PWRSMEVNTMONSTAT0	Section 4.9.4.97
328h	PWRSMEVNTMONSTAT1[(1)]	PWRSMEVNTMONSTAT1	Section 4.9.4.98
32Ch	PWRSMEVNTMONSTAT2[(1)]	PWRSMEVNTMONSTAT2	Section 4.9.4.99
330h	PWRSMWAKESRCSTATCLR0[(1)]	PWRSMWAKESRCSTATCLR0	Section 4.9.4.100
334h	PWRSMWAKESRCSTATCLR1[(1)]	PWRSMWAKESRCSTATCLR1	Section 4.9.4.101
338h	PWRSMWAKESRCSTATCLR2[(1)]	PWRSMWAKESRCSTATCLR2	Section 4.9.4.102
33Ch	ADCBUFCFG1	ADCBUFCFG1	Section 4.9.4.103
340h	ADCBUFCFG2	ADCBUFCFG2	Section 4.9.4.104
344h	ADCBUFCFG3	ADCBUFCFG3	Section 4.9.4.105
348h	ADCBUFCFG4	ADCBUFCFG4	Section 4.9.4.106
34Ch	STCPBISTSMCFG1	STCPBISTSMCFG1	Section 4.9.4.107
350h	STCPBISTSMCFG2	STCPBISTSMCFG2	Section 4.9.4.108
358h	RTI2EVENTCAPTURESEL	RTI2EVENTCAPTURESEL	Section 4.9.4.109
35Ch	DSSMISC5	DSSMISC5	Section 4.9.4.110

Complex bit access types are encoded to fit into small table cells. Table 4-928 shows the codes that are used for access types in this section.

Table 4-632. DSS_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.8.4.1 RTIEVENTCAPTURESEL Register (Offset = 50h) [reset = 0h]

RTIEVENTCAPTURESEL is shown in Figure 4-886 and described in Table 4-929.

Return to Summary Table.

Figure 4-600. RTIEVENTCAPTURESEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2									EVT1							NU1									EVT0							
R-0h									R/W-0h							R-0h									R/W-0h						

Table 4-633. RTIEVENTCAPTURESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	
22-16	EVT1	R/W	0h	Setting the source of interrupt for Counter value capture for RTI1 Event1
15-7	NU1	R	0h	
6-0	EVT0	R/W	0h	Setting the source of interrupt for Counter value capture for RTI1 Event0.

4.8.4.2 CQCFG1 Register (Offset = 6Ch) [reset = 40100000h]

CQCFG1 is shown in Figure 4-887 and described in Table 4-930.

Return to Summary Table.

Figure 4-601. CQCFG1 Register

31	30	29	28	27	26	25	24
NU3	CQ2BASEADDR						
R-0h	R/W-100h						
23	22	21	20	19	18	17	16
CQ2BASEADDR			CQ1BASEADDR				
R/W-100h			R/W-80h				
15	14	13	12	11	10	9	8
CQ1BASEADDR			CQ0BASEADDR				
R/W-80h			R/W-0h				
7	6	5	4	3	2	1	0
CQ0BASEADDR				CQ96BITPACKEN	NU	CQDATAWIDTH	
R/W-0h				R/W-0h	R-0h	R/W-0h	

Table 4-634. CQCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU3	R	0h	
30-22	CQ2BASEADDR	R/W	100h	128-bit Address offset which inidcates the start address for storing CQ0 (ADC/RxIF Saturation Detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
21-13	CQ1BASEADDR	R/W	80h	128-bit Address offset which inidcates the start address for storing CQ0 (Signal Image Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
12-4	CQ0BASEADDR	R/W	0h	128-bit Address offset which inidcates the start address for storing CQ0 (Wide Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
3	CQ96BITPACKEN	R/W	0h	This is used to pack the CQ data into only the LSB 96 bits of each row of the CQ memory. This can be used in 3 channel mode of LVDS where the ADC data and Chirp Params occupy only LSB 96 bits of each memory row.
2	NU	R	0h	
1-0	CQDATAWIDTH	R/W	0h	This is used to appropriately pack the valid CQ data bits in appropriate bits in the CQ memory. 00, 01->Raw 16, 10-->Raw 12, 11-->Raw14

4.8.4.3 TPCCPARSTATCFG Register (Offset = 80h) [reset = 0h]

TPCCPARSTATCFG is shown in Figure 4-888 and described in Table 4-931.

Return to Summary Table.

Figure 4-602. TPCCPARSTATCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU					TPCCPARITYTSTEN	TPCCPARITYEN	TPCCPARITYCLR
R-0h					R/W-0h	R/W-0h	0h
7	6	5	4	3	2	1	0
TPCCPARITYSTAT							
R-0h							

Table 4-635. TPCCPARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	
10	TPCCPARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in TPCC
9	TPCCPARITYEN	R/W	0h	Enable bit for the Parity computation in TPCC
8	TPCCPARITYCLR		0h	Write 0x1 to clear the status. This is a special access type; a write to this field generates a pulse.
7-0	TPCCPARITYSTAT	R	0h	Parity address from TPCC

4.8.4.4 TPTC0WRMPUSTADD0 Register (Offset = 104h) [reset = 0h]

TPTC0WRMPUSTADD0 is shown in Figure 4-889 and described in Table 4-932.

Return to Summary Table.

Figure 4-603. TPTC0WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD0																															
R/W-0h																															

Table 4-636. TPTC0WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC0

4.8.4.5 TPTC0WRMPUSTADD1 Register (Offset = 108h) [reset = 0h]

TPTC0WRMPUSTADD1 is shown in Figure 4-890 and described in Table 4-933.

Return to Summary Table.

Figure 4-604. TPTC0WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD1																															
R/W-0h																															

Table 4-637. TPTC0WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC0

4.8.4.6 TPTC0WRMPUSTADD2 Register (Offset = 10Ch) [reset = 0h]

TPTC0WRMPUSTADD2 is shown in Figure 4-891 and described in Table 4-934.

Return to Summary Table.

Figure 4-605. TPTC0WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD2																															
R/W-0h																															

Table 4-638. TPTC0WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC0

4.8.4.7 TPTC0WRMPUSTADD3 Register (Offset = 110h) [reset = 0h]

TPTC0WRMPUSTADD3 is shown in Figure 4-892 and described in Table 4-935.

Return to Summary Table.

Figure 4-606. TPTC0WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD3																															
R/W-0h																															

Table 4-639. TPTC0WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC0

4.8.4.8 TPTC0WRMPUSTADD4 Register (Offset = 114h) [reset = 0h]

TPTC0WRMPUSTADD4 is shown in Figure 4-893 and described in Table 4-936.

Return to Summary Table.

Figure 4-607. TPTC0WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD4																															
R/W-0h																															

Table 4-640. TPTC0WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC0

4.8.4.9 TPTC0WRMPUSTADD5 Register (Offset = 118h) [reset = 0h]

TPTC0WRMPUSTADD5 is shown in Figure 4-894 and described in Table 4-937.

Return to Summary Table.

Figure 4-608. TPTC0WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD5																															
R/W-0h																															

Table 4-641. TPTC0WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC0

4.8.4.10 TPTC0WRMPUENDADD0 Register (Offset = 124h) [reset = 0h]

TPTC0WRMPUENDADD0 is shown in Figure 4-895 and described in Table 4-938.

Return to Summary Table.

Figure 4-609. TPTC0WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD0																															
R/W-0h																															

Table 4-642. TPTC0WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC0

4.8.4.11 TPTC0WRMPUENDADD1 Register (Offset = 128h) [reset = 0h]

TPTC0WRMPUENDADD1 is shown in Figure 4-896 and described in Table 4-939.

Return to Summary Table.

Figure 4-610. TPTC0WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD1																															
R/W-0h																															

Table 4-643. TPTC0WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC0

4.8.4.12 TPTC0WRMPUENDADD2 Register (Offset = 12Ch) [reset = 0h]

TPTC0WRMPUENDADD2 is shown in Figure 4-897 and described in Table 4-940.

Return to Summary Table.

Figure 4-611. TPTC0WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD2																															
R/W-0h																															

Table 4-644. TPTC0WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC0

4.8.4.13 TPTC0WRMPUENDADD3 Register (Offset = 130h) [reset = 0h]

TPTC0WRMPUENDADD3 is shown in Figure 4-898 and described in Table 4-941.

Return to Summary Table.

Figure 4-612. TPTC0WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD3																															
R/W-0h																															

Table 4-645. TPTC0WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC0

4.8.4.14 TPTC0WRMPUENDADD4 Register (Offset = 134h) [reset = 0h]

TPTC0WRMPUENDADD4 is shown in Figure 4-899 and described in Table 4-942.

Return to Summary Table.

Figure 4-613. TPTC0WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD4																															
R/W-0h																															

Table 4-646. TPTC0WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC0

4.8.4.15 TPTC0WRMPUENDADD5 Register (Offset = 138h) [reset = 0h]

TPTC0WRMPUENDADD5 is shown in Figure 4-900 and described in Table 4-943.

Return to Summary Table.

Figure 4-614. TPTC0WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD5																															
R/W-0h																															

Table 4-647. TPTC0WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC0

4.8.4.16 TPTC0WRMPUERRADD Register (Offset = 144h) [reset = 0h]

TPTC0WRMPUERRADD is shown in Figure 4-901 and described in Table 4-944.

Return to Summary Table.

Figure 4-615. TPTC0WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUERRADD																															
R-0h																															

Table 4-648. TPTC0WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC0

4.8.4.17 TPTC0RDMPUSTADD0 Register (Offset = 148h) [reset = 0h]

TPTC0RDMPUSTADD0 is shown in Figure 4-902 and described in Table 4-945.

Return to Summary Table.

Figure 4-616. TPTC0RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD0																															
R/W-0h																															

Table 4-649. TPTC0RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC0

4.8.4.18 TPTC0RDMPUSTADD1 Register (Offset = 14Ch) [reset = 0h]

TPTC0RDMPUSTADD1 is shown in Figure 4-903 and described in Table 4-946.

Return to Summary Table.

Figure 4-617. TPTC0RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD1																															
R/W-0h																															

Table 4-650. TPTC0RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC0

4.8.4.19 TPTC0RDMPUSTADD2 Register (Offset = 150h) [reset = 0h]

TPTC0RDMPUSTADD2 is shown in Figure 4-904 and described in Table 4-947.

Return to Summary Table.

Figure 4-618. TPTC0RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD2																															
R/W-0h																															

Table 4-651. TPTC0RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC0

4.8.4.20 TPTC0RDMPUSTADD3 Register (Offset = 154h) [reset = 0h]

TPTC0RDMPUSTADD3 is shown in Figure 4-905 and described in Table 4-948.

Return to Summary Table.

Figure 4-619. TPTC0RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD3																															
R/W-0h																															

Table 4-652. TPTC0RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC0

4.8.4.21 TPTC0RDMPUSTADD4 Register (Offset = 158h) [reset = 0h]

TPTC0RDMPUSTADD4 is shown in Figure 4-906 and described in Table 4-949.

Return to Summary Table.

Figure 4-620. TPTC0RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD4																															
R/W-0h																															

Table 4-653. TPTC0RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC0

4.8.4.22 TPTC0RDMPUSTADD5 Register (Offset = 15Ch) [reset = 0h]

TPTC0RDMPUSTADD5 is shown in Figure 4-907 and described in Table 4-950.

Return to Summary Table.

Figure 4-621. TPTC0RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD5																															
R/W-0h																															

Table 4-654. TPTC0RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC0

4.8.4.23 TPTC0RDMPUENDADD0 Register (Offset = 168h) [reset = 0h]

TPTC0RDMPUENDADD0 is shown in Figure 4-908 and described in Table 4-951.

Return to Summary Table.

Figure 4-622. TPTC0RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD0																															
R/W-0h																															

Table 4-655. TPTC0RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC0

4.8.4.24 TPTC0RDMPUENDADD1 Register (Offset = 16Ch) [reset = 0h]

TPTC0RDMPUENDADD1 is shown in Figure 4-909 and described in Table 4-952.

Return to Summary Table.

Figure 4-623. TPTC0RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD1																															
R/W-0h																															

Table 4-656. TPTC0RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC0

4.8.4.25 TPTC0RDMPUENDADD2 Register (Offset = 170h) [reset = 0h]

TPTC0RDMPUENDADD2 is shown in Figure 4-910 and described in Table 4-953.

Return to Summary Table.

Figure 4-624. TPTC0RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD2																															
R/W-0h																															

Table 4-657. TPTC0RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC0

4.8.4.26 TPTC0RDMPUENDADD3 Register (Offset = 174h) [reset = 0h]

TPTC0RDMPUENDADD3 is shown in Figure 4-911 and described in Table 4-954.

Return to Summary Table.

Figure 4-625. TPTC0RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD3																															
R/W-0h																															

Table 4-658. TPTC0RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC0

4.8.4.27 TPTC0RDMPUENDADD4 Register (Offset = 178h) [reset = 0h]

TPTC0RDMPUENDADD4 is shown in Figure 4-912 and described in Table 4-955.

Return to Summary Table.

Figure 4-626. TPTC0RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD4																															
R/W-0h																															

Table 4-659. TPTC0RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC0

4.8.4.28 TPTC0RDMPUENDADD5 Register (Offset = 17Ch) [reset = 0h]

TPTC0RDMPUENDADD5 is shown in Figure 4-913 and described in Table 4-956.

Return to Summary Table.

Figure 4-627. TPTC0RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD5																															
R/W-0h																															

Table 4-660. TPTC0RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC0

4.8.4.29 TPTC0RDMPUERRADD Register (Offset = 188h) [reset = 0h]

TPTC0RDMPUERRADD is shown in Figure 4-914 and described in Table 4-957.

Return to Summary Table.

Figure 4-628. TPTC0RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUERRADD																															
R-0h																															

Table 4-661. TPTC0RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC0

4.8.4.30 TPTC1WRMPUSTADD0 Register (Offset = 18Ch) [reset = 0h]

TPTC1WRMPUSTADD0 is shown in Figure 4-915 and described in Table 4-958.

Return to Summary Table.

Figure 4-629. TPTC1WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD0																															
R/W-0h																															

Table 4-662. TPTC1WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC1

4.8.4.31 TPTC1WRMPUSTADD1 Register (Offset = 190h) [reset = 0h]

TPTC1WRMPUSTADD1 is shown in Figure 4-916 and described in Table 4-959.

Return to Summary Table.

Figure 4-630. TPTC1WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD1																															
R/W-0h																															

Table 4-663. TPTC1WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC1

4.8.4.32 TPTC1WRMPUSTADD2 Register (Offset = 194h) [reset = 0h]

TPTC1WRMPUSTADD2 is shown in Figure 4-917 and described in Table 4-960.

Return to Summary Table.

Figure 4-631. TPTC1WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD2																															
R/W-0h																															

Table 4-664. TPTC1WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC1

4.8.4.33 TPTC1WRMPUSTADD3 Register (Offset = 198h) [reset = 0h]

TPTC1WRMPUSTADD3 is shown in Figure 4-918 and described in Table 4-961.

Return to Summary Table.

Figure 4-632. TPTC1WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD3																															
R/W-0h																															

Table 4-665. TPTC1WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC1

4.8.4.34 TPTC1WRMPUSTADD4 Register (Offset = 19Ch) [reset = 0h]

TPTC1WRMPUSTADD4 is shown in Figure 4-919 and described in Table 4-962.

Return to Summary Table.

Figure 4-633. TPTC1WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD4																															
R/W-0h																															

Table 4-666. TPTC1WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC1

4.8.4.35 TPTC1WRMPUSTADD5 Register (Offset = 1A0h) [reset = 0h]

TPTC1WRMPUSTADD5 is shown in Figure 4-920 and described in Table 4-963.

Return to Summary Table.

Figure 4-634. TPTC1WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD5																															
R/W-0h																															

Table 4-667. TPTC1WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC1

4.8.4.36 TPTC1WRMPUENDADD0 Register (Offset = 1ACh) [reset = 0h]

TPTC1WRMPUENDADD0 is shown in Figure 4-921 and described in Table 4-964.

Return to Summary Table.

Figure 4-635. TPTC1WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD0																															
R/W-0h																															

Table 4-668. TPTC1WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC1

4.8.4.37 TPTC1WRMPUENDADD1 Register (Offset = 1B0h) [reset = 0h]

TPTC1WRMPUENDADD1 is shown in Figure 4-922 and described in Table 4-965.

Return to Summary Table.

Figure 4-636. TPTC1WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD1																															
R/W-0h																															

Table 4-669. TPTC1WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC1

4.8.4.38 TPTC1WRMPUENDADD2 Register (Offset = 1B4h) [reset = 0h]

TPTC1WRMPUENDADD2 is shown in Figure 4-923 and described in Table 4-966.

Return to Summary Table.

Figure 4-637. TPTC1WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD2																															
R/W-0h																															

Table 4-670. TPTC1WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC1

4.8.4.39 TPTC1WRMPUENDADD3 Register (Offset = 1B8h) [reset = 0h]

TPTC1WRMPUENDADD3 is shown in Figure 4-924 and described in Table 4-967.

Return to Summary Table.

Figure 4-638. TPTC1WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD3																															
R/W-0h																															

Table 4-671. TPTC1WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC1

4.8.4.40 TPTC1WRMPUENDADD4 Register (Offset = 1BCh) [reset = 0h]

TPTC1WRMPUENDADD4 is shown in Figure 4-925 and described in Table 4-968.

Return to Summary Table.

Figure 4-639. TPTC1WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD4																															
R/W-0h																															

Table 4-672. TPTC1WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC1

4.8.4.41 TPTC1WRMPUENDADD5 Register (Offset = 1C0h) [reset = 0h]

TPTC1WRMPUENDADD5 is shown in Figure 4-926 and described in Table 4-969.

Return to Summary Table.

Figure 4-640. TPTC1WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD5																															
R/W-0h																															

Table 4-673. TPTC1WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC1

4.8.4.42 TPTC1WRMPUERRADD Register (Offset = 1CCh) [reset = 0h]

TPTC1WRMPUERRADD is shown in Figure 4-927 and described in Table 4-970.

Return to Summary Table.

Figure 4-641. TPTC1WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUERRADD																															
R-0h																															

Table 4-674. TPTC1WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC1

4.8.4.43 TPTC1RDMPUSTADD0 Register (Offset = 1D0h) [reset = 0h]

TPTC1RDMPUSTADD0 is shown in Figure 4-928 and described in Table 4-971.

Return to Summary Table.

Figure 4-642. TPTC1RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD0																															
R/W-0h																															

Table 4-675. TPTC1RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC1

4.8.4.44 TPTC1RDMPUSTADD1 Register (Offset = 1D4h) [reset = 0h]

TPTC1RDMPUSTADD1 is shown in Figure 4-929 and described in Table 4-972.

Return to Summary Table.

Figure 4-643. TPTC1RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD1																															
R/W-0h																															

Table 4-676. TPTC1RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC1

4.8.4.45 TPTC1RDMPUSTADD2 Register (Offset = 1D8h) [reset = 0h]

TPTC1RDMPUSTADD2 is shown in Figure 4-930 and described in Table 4-973.

Return to Summary Table.

Figure 4-644. TPTC1RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD2																															
R/W-0h																															

Table 4-677. TPTC1RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC1

4.8.4.46 TPTC1RDMPUSTADD3 Register (Offset = 1DCh) [reset = 0h]

TPTC1RDMPUSTADD3 is shown in Figure 4-931 and described in Table 4-974.

Return to Summary Table.

Figure 4-645. TPTC1RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD3																															
R/W-0h																															

Table 4-678. TPTC1RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC1

4.8.4.47 TPTC1RDMPUSTADD4 Register (Offset = 1E0h) [reset = 0h]

TPTC1RDMPUSTADD4 is shown in Figure 4-932 and described in Table 4-975.

Return to Summary Table.

Figure 4-646. TPTC1RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD4																															
R/W-0h																															

Table 4-679. TPTC1RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC1

4.8.4.48 TPTC1RDMPUSTADD5 Register (Offset = 1E4h) [reset = 0h]

TPTC1RDMPUSTADD5 is shown in Figure 4-933 and described in Table 4-976.

Return to Summary Table.

Figure 4-647. TPTC1RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD5																															
R/W-0h																															

Table 4-680. TPTC1RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC1

4.8.4.49 TPTC1RDMPUENDADD0 Register (Offset = 1F0h) [reset = 0h]

TPTC1RDMPUENDADD0 is shown in Figure 4-934 and described in Table 4-977.

Return to Summary Table.

Figure 4-648. TPTC1RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD0																															
R/W-0h																															

Table 4-681. TPTC1RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC1

4.8.4.50 TPTC1RDMPUENDADD1 Register (Offset = 1F4h) [reset = 0h]

TPTC1RDMPUENDADD1 is shown in Figure 4-935 and described in Table 4-978.

Return to Summary Table.

Figure 4-649. TPTC1RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD1																															
R/W-0h																															

Table 4-682. TPTC1RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC1

4.8.4.51 TPTC1RDMPUENDADD2 Register (Offset = 1F8h) [reset = 0h]

TPTC1RDMPUENDADD2 is shown in Figure 4-936 and described in Table 4-979.

Return to Summary Table.

Figure 4-650. TPTC1RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD2																															
R/W-0h																															

Table 4-683. TPTC1RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC1

4.8.4.52 TPTC1RDMPUENDADD3 Register (Offset = 1FCh) [reset = 0h]

TPTC1RDMPUENDADD3 is shown in Figure 4-937 and described in Table 4-980.

Return to Summary Table.

Figure 4-651. TPTC1RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD3																															
R/W-0h																															

Table 4-684. TPTC1RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC1

4.8.4.53 TPTC1RDMPUENDADD4 Register (Offset = 200h) [reset = 0h]

TPTC1RDMPUENDADD4 is shown in Figure 4-938 and described in Table 4-981.

Return to Summary Table.

Figure 4-652. TPTC1RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD4																															
R/W-0h																															

Table 4-685. TPTC1RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC1

4.8.4.54 TPTC1RDMPUENDADD5 Register (Offset = 204h) [reset = 0h]

TPTC1RDMPUENDADD5 is shown in Figure 4-939 and described in Table 4-982.

Return to Summary Table.

Figure 4-653. TPTC1RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD5																															
R/W-0h																															

Table 4-686. TPTC1RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC1

4.8.4.55 TPTC1RDMPUERRADD Register (Offset = 210h) [reset = 0h]

TPTC1RDMPUERRADD is shown in Figure 4-940 and described in Table 4-983.

Return to Summary Table.

Figure 4-654. TPTC1RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUERRADD																															
R-0h																															

Table 4-687. TPTC1RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC1

4.8.4.56 TPTCMPUVALIDCFG Register (Offset = 214h) [reset = 0h]

TPTCMPUVALIDCFG is shown in Figure 4-941 and described in Table 4-984.

Return to Summary Table.

Figure 4-655. TPTCMPUVALIDCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPTC1RDMPURNGVLD								TPTC1WRMPURNGVLD							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPURNGVLD								TPTC0WRMPURNGVLD							
R/W-0h								R/W-0h							

Table 4-688. TPTCMPUVALIDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TPTC1RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC1. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
23-16	TPTC1WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC1. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
15-8	TPTC0RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC0. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
7-0	TPTC0WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC0. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled

4.8.4.57 TPTCMPUENCFG Register (Offset = 218h) [reset = 0h]

TPTCMPUENCFG is shown in Figure 4-942 and described in Table 4-985.

Return to Summary Table.

Figure 4-656. TPTCMPUENCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPTC1RDMPUERRCLR	TPTC1WRMPUERRCLR	TPTC0RDMPUERRCLR	TPTC0WRMPUERRCLR	TPTC1RDMPUEN	TPTC1WRMPUEN	TPTC0RDMPUEN	TPTC0WRMPUEN
0h	0h	0h	0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-689. TPTCMPUENCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPTC1RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC1. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
6	TPTC1WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC1. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
5	TPTC0RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC0. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
4	TPTC0WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC0. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
3	TPTC1RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC1. 0 : MPU is disabled 1 : MPU is enabled
2	TPTC1WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC1. 0 : MPU is disabled 1 : MPU is enabled
1	TPTC0RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC0. 0 : MPU is disabled 1 : MPU is enabled
0	TPTC0WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC0. 0 : MPU is disabled 1 : MPU is enabled

4.8.4.58 TESTPATTERNRX1ICFG Register (Offset = 21Ch) [reset = 00010000h]

TESTPATTERNRX1ICFG is shown in Figure 4-943 and described in Table 4-986.

Return to Summary Table.

Figure 4-657. TESTPATTERNRX1ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1IINCR																TSTPATRX1IOFFSET															
R/W-1h																R/W-0h															

Table 4-690. TESTPATTERNRX1ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.8.4.59 TESTPATTERNRX2ICFG Register (Offset = 220h) [reset = 00010000h]

TESTPATTERNRX2ICFG is shown in Figure 4-944 and described in Table 4-987.

Return to Summary Table.

Figure 4-658. TESTPATTERNRX2ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2IINCR																TSTPATRX2IOFFSET															
R/W-1h																R/W-0h															

Table 4-691. TESTPATTERNRX2ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 1.
15-0	TSTPATRX2IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 1.

4.8.4.60 TESTPATTERNRX3ICFG Register (Offset = 224h) [reset = 00010000h]

TESTPATTERNRX3ICFG is shown in Figure 4-945 and described in Table 4-988.

Return to Summary Table.

Figure 4-659. TESTPATTERNRX3ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3IINCR																TSTPATRX3IOFFSET															
R/W-1h																R/W-0h															

Table 4-692. TESTPATTERNRX3ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 2
15-0	TSTPATRX3IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 2

4.8.4.61 TESTPATTERNRX4ICFG Register (Offset = 228h) [reset = 00010000h]

TESTPATTERNRX4ICFG is shown in Figure 4-946 and described in Table 4-989.

Return to Summary Table.

Figure 4-660. TESTPATTERNRX4ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4IINCR																TSTPATRX4IOFFSET															
R/W-1h																R/W-0h															

Table 4-693. TESTPATTERNRX4ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 3
15-0	TSTPATRX4IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 3

4.8.4.62 TESTPATTERNRX1QCFG Register (Offset = 22Ch) [reset = 00010000h]

TESTPATTERNRX1QCFG is shown in Figure 4-947 and described in Table 4-990.

Return to Summary Table.

Figure 4-661. TESTPATTERNRX1QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1QINCR																TSTPATRX1QOFFSET															
R/W-1h																R/W-0h															

Table 4-694. TESTPATTERNRX1QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.8.4.63 TESTPATTERNRX2QCFG Register (Offset = 230h) [reset = 00010000h]

TESTPATTERNRX2QCFG is shown in Figure 4-948 and described in Table 4-991.

Return to Summary Table.

Figure 4-662. TESTPATTERNRX2QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2QINCR																TSTPATRX2QOFFSET															
R/W-1h																R/W-0h															

Table 4-695. TESTPATTERNRX2QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 1.
15-0	TSTPATRX2QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 1.

4.8.4.64 TESTPATTERNRX3QCFG Register (Offset = 234h) [reset = 00010000h]

TESTPATTERNRX3QCFG is shown in Figure 4-949 and described in Table 4-992.

Return to Summary Table.

Figure 4-663. TESTPATTERNRX3QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3QINCR																TSTPATRX3QOFFSET															
R/W-1h																R/W-0h															

Table 4-696. TESTPATTERNRX3QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 2
15-0	TSTPATRX3QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 2

4.8.4.65 TESTPATTERNRX4QCFG Register (Offset = 238h) [reset = 00010000h]

TESTPATTERNRX4QCFG is shown in Figure 4-950 and described in Table 4-993.

Return to Summary Table.

Figure 4-664. TESTPATTERNRX4QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4QINCR																TSTPATRX4QOFFSET															
R/W-1h																R/W-0h															

Table 4-697. TESTPATTERNRX4QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 3
15-0	TSTPATRX4QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 3

4.8.4.66 TESTPATTERNVLDCFG Register (Offset = 23Ch) [reset = 8h]

TESTPATTERNVLDCFG is shown in Figure 4-951 and described in Table 4-994.

Return to Summary Table.

Figure 4-665. TESTPATTERNVLDCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU					TSTPATGENEN		
R-0h					R/W-0h		
7	6	5	4	3	2	1	0
TSTPATVLDCNT							
R/W-8h							

Table 4-698. TESTPATTERNVLDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	
10-8	TSTPATGENEN	R/W	0h	Enable for test pattern generator. This is used to Mux with the functional data from BSS. 000 -->Disable, 111-->Enable, Others are reserved.
7-0	TSTPATVLDCNT	R/W	8h	Number of DSS Interconnect clocks (200 MHz) between successive samples for the test pattern gen.

4.8.4.67 TPCC1PARSTATCFG Register (Offset = 258h) [reset = 0h]

TPCC1PARSTATCFG is shown in Figure 4-952 and described in Table 4-995.

Return to Summary Table.

Figure 4-666. TPCC1PARSTATCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU				TPCC1PARITYTSTEN	TPCC1PARITYEN	TPCC1PARITYCLR	TPCC1PARITYSTAT
R-0h				R/W-0h	R/W-0h	0h	R-0h
7	6	5	4	3	2	1	0
TPCC1PARITYSTAT							
R-0h							

Table 4-699. TPCC1PARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11	TPCC1PARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in TPCC
10	TPCC1PARITYEN	R/W	0h	Enable bit for the Parity computation in TPCC
9	TPCC1PARITYCLR		0h	Clear bit for the Parity error from TPCC Write 0x1 to clear the status. It is a wspecial access type; a write to this field generates a pulse.
8-0	TPCC1PARITYSTAT	R	0h	Parity address from TPCC

4.8.4.68 DMMSWINT1 Register (Offset = 260h) [reset = 0h]

DMMSWINT1 is shown in Figure 4-953 and described in Table 4-996.

Return to Summary Table.

Figure 4-667. DMMSWINT1 Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2	DMMCQWREN	DMMCQPINPONSEL	DMMCPBPMMEMSEL	DMMCPBPMWREN	DMMCPBPMPINPONSEL	DMMADCBUFWREN	DMMADCBUFPINPONSEL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
NU1							
R-0h							
7	6	5	4	3	2	1	0
NU1							
R-0h							

Table 4-700. DMMSWINT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	Not Used
22	DMMCQWREN	R/W	0h	CQ Write Enable from DMM. 0 --> Write to CQ memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from CQ_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMCQPINPONSEL register.
21	DMMCQPINPONSEL	R/W	0h	CQ Ping Pong select for HIL Mode
20	DMMCPBPMMEMSEL	R/W	0h	Select signal for Muxing between HW Registers/Memory for CPBPM data. 0 --> Read access from CPBPM_MEM slave in DSS interconnect will be routed to HW Registers which is populated by DFE/RampGen, 1 --> Read access from CPBPM_MEM slave in DSS interconnect will be routed to appropriate CPBPM memory (Ping/Pong).
19	DMMCPBPMWREN	R/W	0h	CPBPM Write Enable from DMM. 0 --> Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Ping-pong select will come from DMMCPBPMPINPONSEL register.
18	DMMCPBPMPINPONSEL	R/W	0h	CP BPM Ping Pong select for HIL Mode
17	DMMADCBUFWREN	R/W	0h	ADC Buffer Write Enable from DMM. 0 --> Write to ADC BUF memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from ADCBUF_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMADCBUFPINPONSEL register.
16	DMMADCBUFPINPONSEL	R/W	0h	ADC Buffer Ping Pong select for HIL Mode
15-0	NU1	R	0h	Not Used

4.8.4.69 DSSINTRCFG Register (Offset = 270h) [reset = 0h]

DSSINTRCFG is shown in Figure 4-954 and described in Table 4-997.

Return to Summary Table.

Figure 4-668. DSSINTRCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
LGFRAMESTRTINTMUXSEL		PINPONINTMUXSEL		CHIRPAVLINTMUXSEL		FRAMESTRTINTMUXSEL	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 4-701. DSSINTRCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Not Used
7-6	LGFRAMESTRTINTMUXSEL	R/W	0h	[0] - Used to select between DFE and DMM Global Logical Frame Start CFG Bit. 0 --> Select DFE Logical Frame Start, 1--> Select DMM Global Logical Frame start CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 3.
5-4	PINPONINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Ping Pong Switch, 1--> Select DMM Global Ping Pong Switch CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 2.
3-2	CHIRPAVLINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Chirp Available, 1--> Select DMM Global Chirp Available CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 1.
1-0	FRAMESTRTINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Frame Start, 1--> Select DMM Global Frame start CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 0.

4.8.4.70 MPUMSTIDCFG1 Register (Offset = 274h) [reset = 1A191514h]

MPUMSTIDCFG1 is shown in Figure 4-955 and described in Table 4-998.

Return to Summary Table.

Figure 4-669. MPUMSTIDCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUMSTID3								MPUMSTID2							
R/W-1Ah								R/W-19h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPUMSTID1								MPUMSTID0							
R/W-15h								R/W-14h							

Table 4-702. MPUMSTIDCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MPUMSTID3	R/W	1Ah	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID3. Defaule value maps to RS232 Port
23-16	MPUMSTID2	R/W	19h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID2. Defaule value maps to MSS DAP Port
15-8	MPUMSTID1	R/W	15h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID0. Defaule value maps to MSS CR4 Read Port
7-0	MPUMSTID0	R/W	14h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID0. Defaule value maps to MSS CR4 Read Port

4.8.4.71 MPUMSTIDCFG2 Register (Offset = 278h) [reset = 1A191514h]

MPUMSTIDCFG2 is shown in Figure 4-956 and described in Table 4-999.

Return to Summary Table.

Figure 4-670. MPUMSTIDCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUMSTID7								MPUMSTID6							
R/W-1Ah								R/W-19h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPUMSTID5								MPUMSTID4							
R/W-15h								R/W-14h							

Table 4-703. MPUMSTIDCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MPUMSTID7	R/W	1Ah	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID7. Defaule value maps to RS232 Port
23-16	MPUMSTID6	R/W	19h	Allowed MSTID6. Defaule value maps to MSS DAP Port
15-8	MPUMSTID5	R/W	15h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID5 Defaule value maps to MSS CR4 Write Port
7-0	MPUMSTID4	R/W	14h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID4. Defaule value maps to MSS CR4 Read Port

4.8.4.72 MPUMSTIDCFG3 Register (Offset = 27Ch) [reset = FFh]

MPUMSTIDCFG3 is shown in Figure 4-957 and described in Table 4-1000.

Return to Summary Table.

Figure 4-671. MPUMSTIDCFG3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				MPUMSTIDEN	RESERVED	MPUERRCLR	RESERVED
R/W-0h				R/W-0h	R/W-0h	0h	0h
15	14	13	12	11	10	9	8
MPUERRMSTID							
R-0h							
7	6	5	4	3	2	1	0
MPUMSTIDVLD							
R/W-FFh							

Table 4-704. MPUMSTIDCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	MPUMSTIDEN	R/W	0h	Enable control for Master ID based MPU 0 --> Disabled, 1 --> Enabled
18	RESERVED	R/W	0h	Reserved
17	MPUERRCLR		0h	Error clear pulse for Master ID based MPU Write 0x1 to clear the previous error status: it is a wspecial access type; a write to this field generates a pulse.
16	RESERVED		0h	Reserved
15-8	MPUERRMSTID	R	0h	Error status field. Provides the Master ID which is not part of the allowed list which caused an error.
7-0	MPUMSTIDVLD	R/W	FFh	Master ID valid. Each bit corresponds to the MPUMSTID[7:0] 0 : Master ID entry is valid 1 : Master ID entry is not valid and entry does not have access to DSS CFG Space

4.8.4.73 HSRAM1ECCCFG Register (Offset = 280h) [reset = 0h]

HSRAM1ECCCFG is shown in Figure 4-958 and described in Table 4-1001.

Return to Summary Table.

Figure 4-672. HSRAM1ECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	HSRAM1ECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
HSRAM1ECCREPAIREDBIT	HSRAM1ECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
HSRAM1ECCFAULTADDRESS				HSRAM1ECCERRCLR	HSRAM1ECCEN	HSRAM1ECCINITDONE	HSRAM1ECCINIT
R-0h				0h	R/W-0h	R-0h	0h

Table 4-705. HSRAM1ECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	HSRAM1ECCREPAIREDBIT	R	0h	Bit position of the repaired bit in HSRAM1
14-4	HSRAM1ECCFAULTADDRESS	R	0h	ECC Fault address in HSRAM1
3	HSRAM1ECCERRCLR		0h	Clear bit for ECC Error Indication in HSRAM1 Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	HSRAM1ECCEN	R/W	0h	Enable for ECC in HSRAM1
1	HSRAM1ECCINITDONE	R	0h	Done status for ECC Init for HSRAM1
0	HSRAM1ECCINIT		0h	ECC Init For HSRAM1: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.74 DATATRRAMECCCFG Register (Offset = 288h) [reset = 0h]

DATATRRAMECCCFG is shown in Figure 4-959 and described in Table 4-1002.

Return to Summary Table.

Figure 4-673. DATATRRAMECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU			DATATRRAMECCREPAIREDBIT				
R/W-0h			R-0h				
15	14	13	12	11	10	9	8
DATATRRAMECCREPAIREDBIT			DATATRRAMECCFAULTADDRESS				
R-0h			R-0h				
7	6	5	4	3	2	1	0
DATATRRAMECCFAULTADDRESS				DATATRRAMECCERRCLR	DATATRRAMECCEN	DATATRRAMECCINITDONE	DATATRRAMECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-706. DATATRRAMECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU	R	0h	Not Used
20-13	DATATRRAMECCREPAIREDBIT	R	0h	Bit position of the repaired bit in DATATRRAM
12-4	DATATRRAMECCFAULTADDRESS	R	0h	ECC Fault address in DATATRRAM
3	DATATRRAMECCERRCLR	W	0h	Clear bit for ECC Error Indication in DATATRRAM Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	DATATRRAMECCEN	R/W	0h	Enable for ECC in DATATRRAM
1	DATATRRAMECCINITDONE	R	0h	Done status for ECC Init for Data Transfer RAM
0	DATATRRAMECCINIT	W	0h	ECC Init For Data Transfer RAM: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.75 ADCBUFPINGECCCFG Register (Offset = 28Ch) [reset = 0h]

ADCBUFPINGECCCFG is shown in Figure 4-960 and described in Table 4-1003.

Return to Summary Table.

Figure 4-674. ADCBUFPINGECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	ADCBUFPINGECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
ADCBUFPINGECCREPAIREDBIT	ADCBUFPINGECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
ADCBUFPINGECCFAULTADDRESS				ADCBUFPINGECCERRCLR	ADCBUFPINGECCEN	ADCBUFPINGECCINITDONE	ADCBUFPINGECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-707. ADCBUFPINGECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	ADCBUFPINGECCREPAIREDBIT	R	0h	Bit position of the repaired bit in ADC Buffer Ping Memory
14-4	ADCBUFPINGECCFAULTADDRESS	R	0h	ECC Fault address in ADC Buffer Ping Memory
3	ADCBUFPINGECCERRCLR	W	0h	Clear bit for ECC Error Indication in ADC Buffer Ping Memory Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	ADCBUFPINGECCEN	R/W	0h	Enable for ECC in ADC Buffer Ping Memory
1	ADCBUFPINGECCINITDONE	R	0h	Done status for ECC Init for ADC Buffer Ping Memory
0	ADCBUFPINGECCINIT	W	0h	ECC Init For ADC Buffer Ping Memory: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.76 ADCBUFPONGECCCFG Register (Offset = 290h) [reset = 0h]

ADCBUFPONGECCCFG is shown in Figure 4-961 and described in Table 4-1004.

Return to Summary Table.

Figure 4-675. ADCBUFPONGECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	ADCBUFPONGECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
ADCBUFPONGECCREPAIREDBIT	ADCBUFPONGECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
ADCBUFPONGECCFAULTADDRESS				ADCBUFPONGECCERRCLR	ADCBUFPONGECCEN	ADCBUFPONGECCINITDONE	ADCBUFPONGECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-708. ADCBUFPONGECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	ADCBUFPONGECCREPAIREDBIT	R	0h	Bit position of the repaired bit in ADC Buffer Pong Memory
14-4	ADCBUFPONGECCFAULTADDRESS	R	0h	ECC Fault address in ADC Buffer Pong Memory
3	ADCBUFPONGECCERRCLR	W	0h	Clear bit for ECC Error Indication in ADC Buffer Pong Memory Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	ADCBUFPONGECCEN	R/W	0h	Enable for ECC in ADC Buffer Pong Memory
1	ADCBUFPONGECCINITDONE	R	0h	Done status for ECC Init for ADC Buffer Pong Memory
0	ADCBUFPONGECCINIT	W	0h	ECC Init For ADC Buffer Pong Memory: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.77 UMAP0PARITYCFG1 Register (Offset = 29Ch) [reset = 0h]

UMAP0PARITYCFG1 is shown in Figure 4-962 and described in Table 4-1005.

Return to Summary Table.

Figure 4-676. UMAP0PARITYCFG1 Register

31	30	29	28	27	26	25	24
NU						UMAP0BANK23ADDOUT	
R/W-0h						R-0h	
23	22	21	20	19	18	17	16
UMAP0BANK23ADDOUT							
R-0h							
15	14	13	12	11	10	9	8
UMAP0BANK23ADDOUT	UMAP0BANK01ADDOUT						
R-0h	R-0h						
7	6	5	4	3	2	1	0
UMAP0BANK01ADDOUT				UMAP0BANK23ERROUT	UMAP0BANK01ERROUT	UMAP0PARERRCLR	UMAP0PAREN
R-0h				R-0h	R-0h	0h	R/W-0h

Table 4-709. UMAP0PARITYCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU	R/W	0h	Not Used
25-15	UMAP0BANK23ADDOUT	R	0h	Address corresponding to the parity error in Bank2 or Bank3 of UMAP0.
14-4	UMAP0BANK01ADDOUT	R	0h	Address corresponding to the parity error in Bank0 or Bank1 of UMAP0
3	UMAP0BANK23ERROUT	R	0h	Parity Error indication from either Bank2 or Bank3 of UMAP0
2	UMAP0BANK01ERROUT	R	0h	Parity Error indication from either Bank 0 or Bank1 of UMAP0
1	UMAP0PARERRCLR		0h	Clear pulse for all the error status from UMAP0 parity check logic. Self-clearing pulse Write 0x1 to clear the error status
0	UMAP0PAREN	R/W	0h	Enable for UMAP0 Parity Check logic. Assumed to be static. 0 --> Disable 1 --> Enable

4.8.4.78 UMAP0PARITYCFG2 Register (Offset = 2A0h) [reset = 0h]

UMAP0PARITYCFG2 is shown in Figure 4-963 and described in Table 4-1006.

Return to Summary Table.

Figure 4-677. UMAP0PARITYCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP0BANK1BITOUT																UMAP0BANK0BITOUT															
R-0h																R-0h															

Table 4-710. UMAP0PARITYCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP0BANK1BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank1.
15-0	UMAP0BANK0BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank0.

4.8.4.79 UMAP0PARITYCFG3 Register (Offset = 2A4h) [reset = 0h]

UMAP0PARITYCFG3 is shown in Figure 4-964 and described in Table 4-1007.

Return to Summary Table.

Figure 4-678. UMAP0PARITYCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP0BANK3BITOUT																UMAP0BANK2BITOUT															
R-0h																R-0h															

Table 4-711. UMAP0PARITYCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP0BANK3BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank3.
15-0	UMAP0BANK2BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank2.

4.8.4.80 UMAP1PARITYCFG1 Register (Offset = 2A8h) [reset = 0h]

UMAP1PARITYCFG1 is shown in Figure 4-965 and described in Table 4-1008.

Return to Summary Table.

Figure 4-679. UMAP1PARITYCFG1 Register

31	30	29	28	27	26	25	24
NU						UMAP1BANK23ADDOUT	
R-0h						R-0h	
23	22	21	20	19	18	17	16
UMAP1BANK23ADDOUT							
R-0h							
15	14	13	12	11	10	9	8
UMAP1BANK23ADDOUT	UMAP1BANK01ADDOUT						
R-0h	R-0h						
7	6	5	4	3	2	1	0
UMAP1BANK01ADDOUT				UMAP1BANK23ERROUT	UMAP1BANK01ERROUT	UMAP1PARERRCLR	UMAP1PAREN
R-0h				R-0h	R-0h	W-0h	R/W-0h

Table 4-712. UMAP1PARITYCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU	R	0h	Not Used
25-15	UMAP1BANK23ADDOUT	R	0h	Address corresponding to the parity error in Bank2 or Bank3 of UMAP1.
14-4	UMAP1BANK01ADDOUT	R	0h	Address corresponding to the parity error in Bank0 or Bank1 of UMAP1
3	UMAP1BANK23ERROUT	R	0h	Parity Error indication from either Bank2 or Bank3 of UMAP1
2	UMAP1BANK01ERROUT	R	0h	Parity Error indication from either Bank 0 or Bank1 of UMAP1
1	UMAP1PARERRCLR	W	0h	Clear pulse for all the error status from UMAP1 parity check logic. Self-clearing pulse Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP1PAREN	R/W	0h	Enable for UMAP1 Parity Check logic. Assumed to be static. 0 --> Disable 1 --> Enable

4.8.4.81 UMAP1PARITYCFG2 Register (Offset = 2ACh) [reset = 0h]

UMAP1PARITYCFG2 is shown in Figure 4-966 and described in Table 4-1009.

Return to Summary Table.

Figure 4-680. UMAP1PARITYCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP1BANK1BITOUT																UMAP1BANK0BITOUT															
R-0h																R-0h															

Table 4-713. UMAP1PARITYCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP1BANK1BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank1.
15-0	UMAP1BANK0BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank0.

4.8.4.82 UMAP1PARITYCFG3 Register (Offset = 2B0h) [reset = 0h]

UMAP1PARITYCFG3 is shown in Figure 4-967 and described in Table 4-1010.

Return to Summary Table.

Figure 4-681. UMAP1PARITYCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP1BANK3BITOUT																UMAP1BANK2BITOUT															
R-0h																R-0h															

Table 4-714. UMAP1PARITYCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP1BANK3BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank3.
15-0	UMAP1BANK2BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank2.

4.8.4.83 ESMGRP2MASKCFG Register (Offset = 2B4h) [reset = FFFFFFFFh]

ESMGRP2MASKCFG is shown in Figure 4-968 and described in Table 4-1011.

Return to Summary Table.

Figure 4-682. ESMGRP2MASKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGRP2MASK																															
R/W-FFFFFFFFh																															

Table 4-715. ESMGRP2MASKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGRP2MASK	R/W	FFFFFFFFh	Bbit level mask for each of the error signal connected to ESM Group2 input.

4.8.4.84 L2MEMINITCFG1 Register (Offset = 2B8h) [reset = 0h]

L2MEMINITCFG1 is shown in Figure 4-969 and described in Table 4-1012.

Return to Summary Table.

Figure 4-683. L2MEMINITCFG1 Register

31	30	29	28	27	26	25	24
UMAP1BANK3 PARINITDONE	UMAP1BANK2 PARINITDONE	UMAP1BANK1 PARINITDONE	UMAP1BANK0 PARINITDONE	UMAP0BANK3 PARINITDONE	UMAP0BANK2 PARINITDONE	UMAP0BANK1 PARINITDONE	UMAP0BANK0 PARINITDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
UMAP1BANK3 DATAINITDONE	UMAP1BANK2 DATAINITDONE	UMAP1BANK1 DATAINITDONE	UMAP1BANK0 DATAINITDONE	UMAP0BANK3 DATAINITDONE	UMAP0BANK2 DATAINITDONE	UMAP0BANK1 DATAINITDONE	UMAP0BANK0 DATAINITDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
UMAP1BANK3 PARINIT	UMAP1BANK2 PARINIT	UMAP1BANK1 PARINIT	UMAP1BANK0 PARINIT	UMAP0BANK3 PARINIT	UMAP0BANK2 PARINIT	UMAP0BANK1 PARINIT	UMAP0BANK0 PARINIT
0h	0h	0h	0h	0h	0h	0h	0h
7	6	5	4	3	2	1	0
UMAP1BANK3 DATAINIT	UMAP1BANK2 DATAINIT	UMAP1BANK1 DATAINIT	UMAP1BANK0 DATAINIT	UMAP0BANK3 DATAINIT	UMAP0BANK2 DATAINIT	UMAP0BANK1 DATAINIT	UMAP0BANK0 DATAINIT
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-716. L2MEMINITCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	UMAP1BANK3PARINITDONE	R	0h	Init Done status from UMAP1 Bank3 Parity memory
30	UMAP1BANK2PARINITDONE	R	0h	Init Done status from UMAP1 Bank2 Parity memory
29	UMAP1BANK1PARINITDONE	R	0h	Init Done status from UMAP1 Bank1 Parity memory
28	UMAP1BANK0PARINITDONE	R	0h	Init Done status from UMAP1 Bank0 Parity memory
27	UMAP0BANK3PARINITDONE	R	0h	Init Done status from UMAP0 Bank3 Parity memory
26	UMAP0BANK2PARINITDONE	R	0h	Init Done status from UMAP0 Bank2 Parity memory
25	UMAP0BANK1PARINITDONE	R	0h	Init Done status from UMAP0 Bank1 Parity memory
24	UMAP0BANK0PARINITDONE	R	0h	Init Done status from UMAP0 Bank0 Parity memory
23	UMAP1BANK3DATAINITDONE	R	0h	Init Done status from UMAP1 Bank3 Data memory
22	UMAP1BANK2DATAINITDONE	R	0h	Init Done status from UMAP1 Bank2 Data memory
21	UMAP1BANK1DATAINITDONE	R	0h	Init Done status from UMAP1 Bank1 Data memory
20	UMAP1BANK0DATAINITDONE	R	0h	Init Done status from UMAP1 Bank0 Data memory
19	UMAP0BANK3DATAINITDONE	R	0h	Init Done status from UMAP0 Bank3 Data memory
18	UMAP0BANK2DATAINITDONE	R	0h	Init Done status from UMAP0 Bank2 Data memory
17	UMAP0BANK1DATAINITDONE	R	0h	Init Done status from UMAP0 Bank1 Data memory

Table 4-716. L2MEMINITCFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	UMAP0BANK0DATAINITDONE	R	0h	Init Done status from UMAP0 Bank0 Data memory
15	UMAP1BANK3PARINIT		0h	Init trigger for UMAP1 Bank3 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
14	UMAP1BANK2PARINIT		0h	Init trigger for UMAP1 Bank2 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
13	UMAP1BANK1PARINIT		0h	Init trigger for UMAP1 Bank1 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
12	UMAP1BANK0PARINIT		0h	Init trigger for UMAP1 Bank0 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
11	UMAP0BANK3PARINIT		0h	Init trigger for UMAP0 Bank3 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
10	UMAP0BANK2PARINIT		0h	Init trigger for UMAP0 Bank2 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
9	UMAP0BANK1PARINIT		0h	Init trigger for UMAP0 Bank1 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
8	UMAP0BANK0PARINIT		0h	Init trigger for UMAP0 Bank0 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
7	UMAP1BANK3DATAINIT		0h	Init trigger for UMAP1 Bank3 Data memory: it is a wspecial access type; a write to this field generates a pulse.
6	UMAP1BANK2DATAINIT		0h	Init trigger for UMAP1 Bank2 Data memory: it is a wspecial access type; a write to this field generates a pulse.
5	UMAP1BANK1DATAINIT		0h	Init trigger for UMAP1 Bank1 Data memory: it is a wspecial access type; a write to this field generates a pulse.
4	UMAP1BANK0DATAINIT		0h	Init trigger for UMAP1 Bank0 Data memory: it is a wspecial access type; a write to this field generates a pulse.
3	UMAP0BANK3DATAINIT		0h	Init trigger for UMAP0 Bank3 Data memory: it is a wspecial access type; a write to this field generates a pulse.
2	UMAP0BANK2DATAINIT		0h	Init trigger for UMAP0 Bank2 Data memory: it is a wspecial access type; a write to this field generates a pulse.
1	UMAP0BANK1DATAINIT		0h	Init trigger for UMAP0 Bank1 Data memory: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP0BANK0DATAINIT		0h	Init trigger for UMAP0 Bank0 Data memory: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.85 L2MEMINITCFG2 Register (Offset = 2BCh) [reset = 0h]

L2MEMINITCFG2 is shown in Figure 4-970 and described in Table 4-1013.

Return to Summary Table.

Figure 4-684. L2MEMINITCFG2 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
UMAP1BANK1PRAMINITDONE	UMAP1BANK0PRAMINITDONE	UMAP0BANK1PRAMINITDONE	UMAP0BANK0PRAMINITDONE	UMAP1BANK1PRAMINIT	UMAP1BANK0PRAMINIT	UMAP0BANK1PRAMINIT	UMAP0BANK0PRAMINIT
R-0h	R-0h	R-0h	R-0h	W-0h	W-0h	W-0h	W-0h

Table 4-717. L2MEMINITCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Not Used
7	UMAP1BANK1PRAMINITDONE	R	0h	Init Done Status for UMAP1 Bank1 PRAM memory
6	UMAP1BANK0PRAMINITDONE	R	0h	Init Done Status for UMAP1 Bank0 PRAM memory
5	UMAP0BANK1PRAMINITDONE	R	0h	Init Done Status for UMAP0 Bank1 PRAM memory
4	UMAP0BANK0PRAMINITDONE	R	0h	Init Done Status for UMAP0 Bank0 PRAM memory
3	UMAP1BANK1PRAMINIT	W	0h	Init trigger for UMAP1 Bank1 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
2	UMAP1BANK0PRAMINIT	W	0h	Init trigger for UMAP1 Bank0 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
1	UMAP0BANK1PRAMINIT	W	0h	Init trigger for UMAP0 Bank1 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP0BANK0PRAMINIT	W	0h	Init trigger for UMAP0 Bank0 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.86 GEMRSTCAUSE Register (Offset = 2C0h) [reset = 00010101h]

GEMRSTCAUSE is shown in Figure 4-971 and described in Table 4-1014.

Return to Summary Table.

Figure 4-685. GEMRSTCAUSE Register

31	30	29	28	27	26	25	24
NU2							GEMRSTCAUSECLR
R-0h							0h
23	22	21	20	19	18	17	16
GEMPORCAUSE							
R-1h							
15	14	13	12	11	10	9	8
GEMGRSTCAUSE							
R-1h							
7	6	5	4	3	2	1	0
GEMLRSTCAUSE							
R-1h							

Table 4-718. GEMRSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU2	R	0h	
24	GEMRSTCAUSECLR		0h	Write 0x1 to clear the reset cause register for any previous resets: it is a wspecial access type; a write to this field generates a pulse.
23-16	GEMPORCAUSE	R	1h	DSP POR reset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMPORZ Bit 3 : Reset from Power FSM Bit 4 : Reset from STC FSM
15-8	GEMGRSTCAUSE	R	1h	DSP Greset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMGRSTN Bit 3 : Reset from Power FSM Bit 4 : Reset from STC FSM
7-0	GEMLRSTCAUSE	R	1h	DSP Lreset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMLRSTN Bit 3 : Reset from Debugss Bit 4 : Reset from Power FSM Bit 5 : Reset from STC FSM

4.8.4.87 GEMPWRSMCFG4 Register (Offset = 2CCh) [reset = 00060000h]

GEMPWRSMCFG4 is shown in Figure 4-972 and described in Table 4-1015.

Return to Summary Table.

Figure 4-686. GEMPWRSMCFG4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					GEMEVENTMASK	PWRSMLRSTHALT	PWRSMSLEEPTRIG
R-0h					R/W-1h	R/W-1h	0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 4-719. GEMPWRSMCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18	GEMEVENTMASK	R/W	1h	Mask bit for events going to DSP. When this bit is set (during GEM sleep/power down mode), the events are monitored outside and will be available for DSP to read and clear them once GEM wakes-up. The monitored events can be read from PWRSMEVNTMONSTATx registers.
17	PWRSMLRSTHALT	R/W	1h	Signal to halt DSP Power cycle state machine before de-asserting LRST of DSP. This is used during code download for the first time power up.
16	PWRSMSLEEPTRIG		0h	Sleep mode trigger for DSP power down state machine. This is honoured only when DSP is in GEM_ON state: it is a wspecial access type; a write to this field generates a pulse.
15-0	RESERVED	R	0h	Reserved

4.8.4.88 PWRSMWAKEMASK0 Register (Offset = 2D4h) [reset = FFFFFFFFh]

PWRSMWAKEMASK0 is shown in Figure 4-973 and described in Table 4-1016.

Return to Summary Table.

Figure 4-687. PWRSMWAKEMASK0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK0																															
R/W-FFFFFFFFh																															

Table 4-720. PWRSMWAKEMASK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK0	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [31:0] 1 --> Masked, 0 --> Unmasked.

4.8.4.89 PWRSMWAKEMASK1 Register (Offset = 2D8h) [reset = FFFFFFFFh]

PWRSMWAKEMASK1 is shown in Figure 4-974 and described in Table 4-1017.

Return to Summary Table.

Figure 4-688. PWRSMWAKEMASK1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK1																															
R/W-FFFFFFFFh																															

Table 4-721. PWRSMWAKEMASK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK1	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [63:32] 1 --> Masked, 0 --> Unmasked.

4.8.4.90 PWRSMWAKEMASK2 Register (Offset = 2DCh) [reset = FFFFFFFFh]

PWRSMWAKEMASK2 is shown in Figure 4-975 and described in Table 4-1018.

Return to Summary Table.

Figure 4-689. PWRSMWAKEMASK2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK2																															
R/W-FFFFFFFFh																															

Table 4-722. PWRSMWAKEMASK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK2	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [95:64] 1 --> Masked, 0 --> Unmasked.

4.8.4.91 PWRSMMISEVTMASK0 Register (Offset = 2E0h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK0 is shown in Figure 4-976 and described in Table 4-1019.

Return to Summary Table.

Figure 4-690. PWRSMMISEVTMASK0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK0																															
R/W-FFFFFFFFh																															

Table 4-723. PWRSMMISEVTMASK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK0	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[31:0] 1 --> Masked, 0 --> Unmasked.

4.8.4.92 PWRSMMISEVTMASK1 Register (Offset = 2E4h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK1 is shown in Figure 4-977 and described in Table 4-1020.

Return to Summary Table.

Figure 4-691. PWRSMMISEVTMASK1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK1																															
R/W-FFFFFFFFh																															

Table 4-724. PWRSMMISEVTMASK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK1	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[63:32] 1 --> Masked, 0 --> Unmasked.

4.8.4.93 PWRSMMISEVTMASK2 Register (Offset = 2E8h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK2 is shown in Figure 4-978 and described in Table 4-1021.

Return to Summary Table.

Figure 4-692. PWRSMMISEVTMASK2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK2																															
R/W-FFFFFFFFh																															

Table 4-725. PWRSMMISEVTMASK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK2	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[95:64] 1 --> Masked, 0 --> Unmasked.

4.8.4.94 PWRSMWAKESRCSTAT0 Register (Offset = 2ECh) [reset = 0h]

PWRSMWAKESRCSTAT0 is shown in Figure 4-979 and described in Table 4-1022.

Return to Summary Table.

Figure 4-693. PWRSMWAKESRCSTAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT0																															
R-0h																															

Table 4-726. PWRSMWAKESRCSTAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT0	R	0h	Wakeup source status bits [31:0]

4.8.4.95 PWRSMWAKESRCSTAT1 Register (Offset = 2F0h) [reset = 0h]

PWRSMWAKESRCSTAT1 is shown in Figure 4-980 and described in Table 4-1023.

Return to Summary Table.

Figure 4-694. PWRSMWAKESRCSTAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT1																															
R-0h																															

Table 4-727. PWRSMWAKESRCSTAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT1	R	0h	Wakeup source status bits [63:32]

4.8.4.96 PWRSMWAKESRCSTAT2 Register (Offset = 320h) [reset = 0h]

PWRSMWAKESRCSTAT2 is shown in Figure 4-981 and described in Table 4-1024.

Return to Summary Table.

Figure 4-695. PWRSMWAKESRCSTAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT2																															
R-0h																															

Table 4-728. PWRSMWAKESRCSTAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT2	R	0h	Wakeup source status bits [95:64]

4.8.4.97 PWRSMEVNTMONSTAT0 Register (Offset = 324h) [reset = 0h]

PWRSMEVNTMONSTAT0 is shown in Figure 4-982 and described in Table 4-1025.

Return to Summary Table.

Figure 4-696. PWRSMEVNTMONSTAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT0																															
R-0h																															

Table 4-729. PWRSMEVNTMONSTAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT0	R	0h	Missed events monitor status bits [31:0]. This is monitored when the events going to DSP are masked by GEMEVENTMASK register.

4.8.4.98 PWRSMEVNTMONSTAT1 Register (Offset = 328h) [reset = 0h]

PWRSMEVNTMONSTAT1 is shown in Figure 4-983 and described in Table 4-1026.

Return to Summary Table.

Figure 4-697. PWRSMEVNTMONSTAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT1																															
R-0h																															

Table 4-730. PWRSMEVNTMONSTAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT1	R	0h	Missed events monitor status bits [63:32]

4.8.4.99 PWRSMEVNTMONSTAT2 Register (Offset = 32Ch) [reset = 0h]

PWRSMEVNTMONSTAT2 is shown in Figure 4-984 and described in Table 4-1027.

Return to Summary Table.

Figure 4-698. PWRSMEVNTMONSTAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT2																															
R-0h																															

Table 4-731. PWRSMEVNTMONSTAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT2	R	0h	Missed events monitor status bits [95:64]

4.8.4.100 PWRSMWAKESRCSTATCLR0 Register (Offset = 330h) [reset = 0h]

PWRSMWAKESRCSTATCLR0 is shown in Figure 4-985 and described in Table 4-1028.

Return to Summary Table.

Figure 4-699. PWRSMWAKESRCSTATCLR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR0																															
0h																															

Table 4-732. PWRSMWAKESRCSTATCLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR0		0h	Clear bit for wakeup source status bits [31:0]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.101 PWRSMWAKESRCSTATCLR1 Register (Offset = 334h) [reset = 0h]

PWRSMWAKESRCSTATCLR1 is shown in Figure 4-986 and described in Table 4-1029.

Return to Summary Table.

Figure 4-700. PWRSMWAKESRCSTATCLR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR1																															
0h																															

Table 4-733. PWRSMWAKESRCSTATCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR1		0h	Clear bit for wakeup source status bits [63:32]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.102 PWRSMWAKESRCSTATCLR2 Register (Offset = 338h) [reset = 0h]

PWRSMWAKESRCSTATCLR2 is shown in Figure 4-987 and described in Table 4-1030.

Return to Summary Table.

Figure 4-701. PWRSMWAKESRCSTATCLR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR2																															
0h																															

Table 4-734. PWRSMWAKESRCSTATCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR2		0h	Clear bit for wakeup source status bits [95:64]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.8.4.103 ADCBUFCFG1 Register (Offset = 33Ch) [reset = 00010000h]

ADCBUFCFG1 is shown in Figure 4-988 and described in Table 4-1031.

Return to Summary Table.

Figure 4-702. ADCBUFCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
ADCBUFCONTSTOPPL	ADCBUFCONTSTRTPL	ADCBUFCONTMODEEN	ADCBUFWRITEMODE	RESERVED		RX3EN	RX2EN
W-0h	W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RX1EN	RX0EN	ADCBUFIQSWAP	RESERVED		ADCBUFREALONLYMODE	RESERVED	
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 4-735. ADCBUFCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved
15	ADCBUFCONTSTOPPL	W	0h	Stop Pulse for Continuous mode. The data capture will stop once this register is set. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
14	ADCBUFCONTSTRTPL	W	0h	Start Pulse for Continuous mode. The data capture will start from Address 0 once this register is set. All the other configurations like Enable, Sample Count are expected to be programmed before this pulse. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
13	ADCBUFCONTMODEEN	R/W	0h	Continuous mode enable for ADC Buffer. This is set when a fixed number of samples have to be stored in Ping/Pong and not depend on Chirp time-lines (Eg: Analog Lab characterization to stream out continuous data from DFE). Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
12	ADCBUFWRITEMODE	R/W	0h	This needs to be programmed to 0x1 in 16xx 0 --> Interleaved, 1 --> Non-interleaved
11-10	RESERVED	R/W	0h	Reserved
9	RX3EN	R/W	0h	Enable for Rx3 write
8	RX2EN	R/W	0h	Enable for Rx2 write
7	RX1EN	R/W	0h	Enable for Rx1 write
6	RX0EN	R/W	0h	Enable for Rx0 write
5	ADCBUFIQSWAP	R/W	0h	0 --> I is stored in LSB and Q is stored in MSB 1 --> Q is stored in LSB and I is stored in MSB
4-3	RESERVED	R/W	0h	Reserved
2	ADCBUFREALONLYMODE	R/W	0h	0-->Complex Data mode, 1-->Real data mode
1-0	RESERVED	R/W	0h	Reserved

4.8.4.104 ADCBUFCFG2 Register (Offset = 340h) [reset = 02000000h]

ADCBUFCFG2 is shown in Figure 4-989 and described in Table 4-1032.

Return to Summary Table.

Figure 4-703. ADCBUFCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					ADCBUFADDRX1										
R-0h					R/W-200h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					ADCBUFADDRX0										
R-0h					R/W-0h										

Table 4-736. ADCBUFCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	
26-16	ADCBUFADDRX1	R/W	200h	128 bit Address offset to be added to the internal address pointer for Rx1 writes in Non-interleaved mode.
15-11	NU1	R	0h	
10-0	ADCBUFADDRX0	R/W	0h	128 bit Address offset to be added to the internal address pointer for Rx0 writes in Non-interleaved mode.

4.8.4.105 ADCBUFCFG3 Register (Offset = 344h) [reset = 06000400h]

ADCBUFCFG3 is shown in Figure 4-990 and described in Table 4-1033.

Return to Summary Table.

Figure 4-704. ADCBUFCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					ADCBUFADDRX3										
R-0h					R/W-600h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					ADCBUFADDRX2										
R-0h					R/W-400h										

Table 4-737. ADCBUFCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	
26-16	ADCBUFADDRX3	R/W	600h	128 bit Address offset to be added to the internal address pointer for Rx3 writes in Non-interleaved mode.
15-11	NU1	R	0h	
10-0	ADCBUFADDRX2	R/W	400h	128 bit Address offset to be added to the internal address pointer for Rx2 writes in Non-interleaved mode.

4.8.4.106 ADCBUFCFG4 Register (Offset = 348h) [reset = 400h]

ADCBUFCFG4 is shown in Figure 4-991 and described in Table 4-1034.

Return to Summary Table.

Figure 4-705. ADCBUFCFG4 Register

31	30	29	28	27	26	25	24
RESERVED						ADCBUFNUMCHRPPONG	
R-0h						R/W-0h	
23	22	21	20	19	18	17	16
ADCBUFNUMCHRPPONG			ADCBUFNUMCHRPPING				
R/W-0h			R/W-0h				
15	14	13	12	11	10	9	8
ADCBUFSAMPCNT							
R/W-400h							
7	6	5	4	3	2	1	0
ADCBUFSAMPCNT							
R/W-400h							

Table 4-738. ADCBUFCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-21	ADCBUFNUMCHRPPONG	R/W	0h	Number of chirps to be stored in Pong buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Ping
20-16	ADCBUFNUMCHRPPING	R/W	0h	Number of chirps to be stored in Ping buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Pong
15-0	ADCBUFSAMPCNT	R/W	400h	No of samples to store in each Ping and Pong register in continuous mode of ADC Buffer. In real only mode this refers to the number of real samples and in complex mode, this refers to number of complex samples. This refers to the number of samples per channel. This counter increments once for every new sample from DFE (as long as 1 or more channels are enabled). The max allowed value varies depending on other configurations (No of channels enabled and real/complex data). Continous mode is expected to be only used for CZ and ADC Buffer Testpattern mode

4.8.4.107 STCPBISTSMCFG1 Register (Offset = 34Ch) [reset = 18h]

STCPBISTSMCFG1 is shown in Figure 4-992 and described in Table 4-1035.

Return to Summary Table.

Figure 4-706. STCPBISTSMCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED			PBISTTESTSTATCLR	PBISTTESTSTAT		STCPBISTSMSTATE	
R-0h			W-0h	R-0h		R-0h	
15	14	13	12	11	10	9	8
STCPBISTSMSTATE				RESERVED			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED			STCPBISTCKSTPACKMASK	STCPBISTLRSTDASRTHALT	STCPBISTSMTRIG	STCPBISTEN	
R/W-0h			R/W-1h	R/W-1h	W-0h	R/W-0h	

Table 4-739. STCPBISTSMCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	Reserved
20	PBISTTESTSTATCLR	W	0h	Clear bit for PBIST Status: this is a wspecial access type - awrite to this field generates a pulse.
19-18	PBISTTESTSTAT	R	0h	PBIST status from GEM. [0] - Fail Indication [1] - Done indication
17-12	STCPBISTSMSTATE	R	0h	Current state of STC PBIST state machine
11-5	RESERVED	R/W	0h	Reserved
4	STCPBISTCKSTPACKMASK	R/W	1h	Mask bit for ignoring the clock stop ack from GEM. This will be used for ignoring clock stop ack during boot-up. 1 --> Ignore clock stop ack from GEM. 0 --> Wait for clock stop ack from GEM after giving clock stop request.
3	STCPBISTLRSTDASRTHALT	R/W	1h	Configuration to halt the state machine before the final de-assertion of LRST to enable program download. 1 --> Halt, 0 --> Proceed.
2	STCPBISTSMTRIG	W	0h	Trigger pulse for the STC PBIST state machine. This is a self-clearing pulse.
1-0	STCPBISTEN	R/W	0h	Enable for PBIST and STC. 00 - Reserved, 01 --> STC only 10 --> PBIST only 11 --> PBIST followed by STC.

4.8.4.108 STCPBISTSMCFG2 Register (Offset = 350h) [reset = 2410h]

STCPBISTSMCFG2 is shown in Figure 4-993 and described in Table 4-1036.

Return to Summary Table.

Figure 4-707. STCPBISTSMCFG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							BCK2BCKSTCEN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED		GEMPBISTROMCLKSEL		GEMTMODEVLCTASRTCNT			
R-0h		R/W-2h		R/W-10h			
7	6	5	4	3	2	1	0
GEMTMODEVLCTASRTCNT		GEMTMODEVLCTDASRTCNT					
R/W-10h		R/W-10h					

Table 4-740. STCPBISTSMCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	BCK2BCKSTCEN	R/W	0h	Enables back to Back STC.Needs to be set to 1 for self test
15-14	RESERVED	R	0h	Reserved
13-12	GEMPBISTROMCLKSEL	R/W	2h	Pbist_rom_clk_div_sel --> It is used to select the PBIST rom clock frequency . 2b00 : Div 1 (600Mhz) 2b01 : Div 2 (300Mhz) 2b10 : Div 3 (200Mhz) 2b11 : Div 4 (150Mhz)
11-6	GEMTMODEVLCTASRTCNT	R/W	10h	No of clocks (in terms of 200 MHz DSPSS Bus clock) after asserting GEM TMODE VLCT signal before proceeding to the next state. Max allowed value is 31.
5-0	GEMTMODEVLCTDASRTCNT	R/W	10h	No of clocks (in terms of 200 MHz DSPSS Bus clock) after De-asserting GEM TMODE VLCT signal before proceeding to the next state. Max allowed value is 31.

4.8.4.109 RTI2EVENTCAPTURESEL Register (Offset = 358h) [reset = 0h]

RTI2EVENTCAPTURESEL is shown in Figure 4-994 and described in Table 4-1037.

Return to Summary Table.

Figure 4-708. RTI2EVENTCAPTURESEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2									RTI2EVT1						
R-0h									R/W-0h						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1									RTI2EVT0						
R-0h									R/W-0h						

Table 4-741. RTI2EVENTCAPTURESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	
22-16	RTI2EVT1	R/W	0h	Used to Select the event to be captured for RTI2 Event1.
15-7	NU1	R	0h	
6-0	RTI2EVT0	R/W	0h	Used to Select the event to be captured for RTI2 Event0.

4.8.4.110 DSSMISC5 Register (Offset = 35Ch) [reset = 0h]

DSSMISC5 is shown in Figure 4-995 and described in Table 4-1038.

Return to Summary Table.

Figure 4-709. DSSMISC5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPCC1PARMEMINITDONE	TPCC0PARMEMINITDONE	TPCC1PARMEMINIT	TPCC0PARMEMINIT	CPBPMPIPOSELVAL	CPBPMPIPOSELCNT	CQPIPOSELVAL	CQPIPOSELCNT
R-0h	R-0h	W-0h	W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-742. DSSMISC5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPCC1PARMEMINITDONE	R	0h	Mem init done status for the TPCC1 parity memory
6	TPCC0PARMEMINITDONE	R	0h	Mem init done status for the TPCC0 parity memory
5	TPCC1PARMEMINIT	W	0h	Mem init for the TPCC1 parity memory: this is a wspecial access type, a write to this field generates a pulse.
4	TPCC0PARMEMINIT	W	0h	Mem init for the TPCC0 parity memory: this is a wspecial access type, a write to this field generates a pulse.
3	CPBPMPIPOSELVAL	R/W	0h	Ping pong select override value for CPBPM Memory. 1 --> Read access from CPBPM_MEM Slave of DSS Interconnect will be routed to ping memory and write access from CPBPM_W write will be routed to pong memory. 0 --> Read access from CPBPM_MEM Slave of DSS Interconnect will be routed to pong memory and write access from CPBPM_W write will be routed to ping memory.
2	CPBPMPIPOSELCNT	R/W	0h	Ping pong select override control for CPBPM Memory. 0 --> Ping-pong select comes from HW FSM (same as the ping-pong select for ADC Buffer)/DMMCPBPMPINPONSEL 1 --> Ping pong select for CPBPM memory is taken from SW register (CPBPMPIPOSELVAL)
1	CQPIPOSELVAL	R/W	0h	Ping pong select override value for CQ Memory. 1 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to ping memory and write access from CQ_W/DFE write will be routed to pong memory. 0 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to pong memory and write access from CQ_W/DFE write will be routed to ping memory.
0	CQPIPOSELCNT	R/W	0h	Ping pong select override control for CQ Memory. 0 --> Ping-pong select comes from HW FSM (same as the ping-pong select for ADC Buffer)/DMMCQPINPONSEL 1 --> Ping pong select for CQ memory is taken from SW register (CQPIPOSELVAL)

4.8.5 DSS_REG2 Registers

Table 4-1039 lists the memory-mapped registers for the DSS_REG2. All register offset addresses not listed in Table 4-1039 should be considered as reserved locations and the register contents should not be modified.

Table 4-743. DSS_REG2 Registers

Offset	Acronym	Register Name	Section
100h	TPTC2WRMPUSTADD0	TPTC2WRMPUSTADD0	Section 4.9.5.1
104h	TPTC2WRMPUSTADD1	TPTC2WRMPUSTADD1	Section 4.9.5.2
108h	TPTC2WRMPUSTADD2	TPTC2WRMPUSTADD2	Section 4.9.5.3
10Ch	TPTC2WRMPUSTADD3	TPTC2WRMPUSTADD3	Section 4.9.5.4
110h	TPTC2WRMPUSTADD4	TPTC2WRMPUSTADD4	Section 4.9.5.5
114h	TPTC2WRMPUSTADD5	TPTC2WRMPUSTADD5	Section 4.9.5.6
120h	TPTC2WRMPUENDADD0	TPTC2WRMPUENDADD0	Section 4.9.5.7
124h	TPTC2WRMPUENDADD1	TPTC2WRMPUENDADD1	Section 4.9.5.8
128h	TPTC2WRMPUENDADD2	TPTC2WRMPUENDADD2	Section 4.9.5.9
12Ch	TPTC2WRMPUENDADD3	TPTC2WRMPUENDADD3	Section 4.9.5.10
130h	TPTC2WRMPUENDADD4	TPTC2WRMPUENDADD4	Section 4.9.5.11
134h	TPTC2WRMPUENDADD5	TPTC2WRMPUENDADD5	Section 4.9.5.12
140h	TPTC2WRMPUERRADD	TPTC2WRMPUERRADD	Section 4.9.5.13
148h	TPTC2RDMPUSTADD0	TPTC2RDMPUSTADD0	Section 4.9.5.14
14Ch	TPTC2RDMPUSTADD1	TPTC2RDMPUSTADD1	Section 4.9.5.15
150h	TPTC2RDMPUSTADD2	TPTC2RDMPUSTADD2	Section 4.9.5.16
154h	TPTC2RDMPUSTADD3	TPTC2RDMPUSTADD3	Section 4.9.5.17
158h	TPTC2RDMPUSTADD4	TPTC2RDMPUSTADD4	Section 4.9.5.18
15Ch	TPTC2RDMPUSTADD5	TPTC2RDMPUSTADD5	Section 4.9.5.19
168h	TPTC2RDMPUENDADD0	TPTC2RDMPUENDADD0	Section 4.9.5.20
16Ch	TPTC2RDMPUENDADD1	TPTC2RDMPUENDADD1	Section 4.9.5.21
170h	TPTC2RDMPUENDADD2	TPTC2RDMPUENDADD2	Section 4.9.5.22
174h	TPTC2RDMPUENDADD3	TPTC2RDMPUENDADD3	Section 4.9.5.23
178h	TPTC2RDMPUENDADD4	TPTC2RDMPUENDADD4	Section 4.9.5.24
17Ch	TPTC2RDMPUENDADD5	TPTC2RDMPUENDADD5	Section 4.9.5.25
188h	TPTC2RDMPUERRADD	TPTC2RDMPUERRADD	Section 4.9.5.26
18Ch	TPTC3WRMPUSTADD0	TPTC3WRMPUSTADD0	Section 4.9.5.27
190h	TPTC3WRMPUSTADD1	TPTC3WRMPUSTADD1	Section 4.9.5.28
194h	TPTC3WRMPUSTADD2	TPTC3WRMPUSTADD2	Section 4.9.5.29
198h	TPTC3WRMPUSTADD3	TPTC3WRMPUSTADD3	Section 4.9.5.30
19Ch	TPTC3WRMPUSTADD4	TPTC3WRMPUSTADD4	Section 4.9.5.31
1A0h	TPTC3WRMPUSTADD5	TPTC3WRMPUSTADD5	Section 4.9.5.32
1ACh	TPTC3WRMPUENDADD0	TPTC3WRMPUENDADD0	Section 4.9.5.33
1B0h	TPTC3WRMPUENDADD1	TPTC3WRMPUENDADD1	Section 4.9.5.34
1B4h	TPTC3WRMPUENDADD2	TPTC3WRMPUENDADD2	Section 4.9.5.35
1B8h	TPTC3WRMPUENDADD3	TPTC3WRMPUENDADD3	Section 4.9.5.36
1BCh	TPTC3WRMPUENDADD4	TPTC3WRMPUENDADD4	Section 4.9.5.37
1C0h	TPTC3WRMPUENDADD5	TPTC3WRMPUENDADD5	Section 4.9.5.38
1CCh	TPTC3WRMPUERRADD	TPTC3WRMPUERRADD	Section 4.9.5.39
1D0h	TPTC3RDMPUSTADD0	TPTC3RDMPUSTADD0	Section 4.9.5.40
1D4h	TPTC3RDMPUSTADD1	TPTC3RDMPUSTADD1	Section 4.9.5.41
1D8h	TPTC3RDMPUSTADD2	TPTC3RDMPUSTADD2	Section 4.9.5.42
1DCh	TPTC3RDMPUSTADD3	TPTC3RDMPUSTADD3	Section 4.9.5.43

Table 4-743. DSS_REG2 Registers (continued)

Offset	Acronym	Register Name	Section
1E0h	TPTC3RDMPUSTADD4	TPTC3RDMPUSTADD4	Section 4.9.5.44
1E4h	TPTC3RDMPUSTADD5	TPTC3RDMPUSTADD5	Section 4.9.5.45
1F0h	TPTC3RDMPUENDADD0	TPTC3RDMPUENDADD0	Section 4.9.5.46
1F4h	TPTC3RDMPUENDADD1	TPTC3RDMPUENDADD1	Section 4.9.5.47
1F8h	TPTC3RDMPUENDADD2	TPTC3RDMPUENDADD2	Section 4.9.5.48
1FCh	TPTC3RDMPUENDADD3	TPTC3RDMPUENDADD3	Section 4.9.5.49
200h	TPTC3RDMPUENDADD4	TPTC3RDMPUENDADD4	Section 4.9.5.50
204h	TPTC3RDMPUENDADD5	TPTC3RDMPUENDADD5	Section 4.9.5.51
210h	TPTC3RDMPUERRADD	TPTC3RDMPUERRADD	Section 4.9.5.52
214h	TPTCMPUVALIDCFG2	TPTCMPUVALIDCFG2	Section 4.9.5.53
218h	TPTCMPUENCFG2	TPTCMPUENCFG2	Section 4.9.5.54
268h	L3ECCCFG1	L3ECCCFG1	Section 4.9.5.55
26Ch	L3ECCCFG2	L3ECCCFG2	Section 4.9.5.56
270h	DSS2MSSSWIRQ		Section 4.9.5.57

Complex bit access types are encoded to fit into small table cells. Table 4-1040 shows the codes that are used for access types in this section.

Table 4-744. DSS_REG2 Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.8.5.1 TPTC2WRMPUSTADD0 Register (Offset = 100h) [reset = 0h]

TPTC2WRMPUSTADD0 is shown in Figure 4-996 and described in Table 4-1041.

Return to Summary Table.

Figure 4-710. TPTC2WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD0																															
R/W-0h																															

Table 4-745. TPTC2WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC2

4.8.5.2 TPTC2WRMPUSTADD1 Register (Offset = 104h) [reset = 0h]

TPTC2WRMPUSTADD1 is shown in Figure 4-997 and described in Table 4-1042.

Return to Summary Table.

Figure 4-711. TPTC2WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD1																															
R/W-0h																															

Table 4-746. TPTC2WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC2

4.8.5.3 TPTC2WRMPUSTADD2 Register (Offset = 108h) [reset = 0h]

TPTC2WRMPUSTADD2 is shown in Figure 4-998 and described in Table 4-1043.

Return to Summary Table.

Figure 4-712. TPTC2WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD2																															
R/W-0h																															

Table 4-747. TPTC2WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC2

4.8.5.4 TPTC2WRMPUSTADD3 Register (Offset = 10Ch) [reset = 0h]

TPTC2WRMPUSTADD3 is shown in Figure 4-999 and described in Table 4-1044.

Return to Summary Table.

Figure 4-713. TPTC2WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD3																															
R/W-0h																															

Table 4-748. TPTC2WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC2

4.8.5.5 TPTC2WRMPUSTADD4 Register (Offset = 110h) [reset = 0h]

TPTC2WRMPUSTADD4 is shown in Figure 4-1000 and described in Table 4-1045.

Return to Summary Table.

Figure 4-714. TPTC2WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD4																															
R/W-0h																															

Table 4-749. TPTC2WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC2

4.8.5.6 TPTC2WRMPUSTADD5 Register (Offset = 114h) [reset = 0h]

TPTC2WRMPUSTADD5 is shown in Figure 4-1001 and described in Table 4-1046.

Return to Summary Table.

Figure 4-715. TPTC2WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD5																															
R/W-0h																															

Table 4-750. TPTC2WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC2

4.8.5.7 TPTC2WRMPUENDADD0 Register (Offset = 120h) [reset = 0h]

TPTC2WRMPUENDADD0 is shown in Figure 4-1002 and described in Table 4-1047.

Return to Summary Table.

Figure 4-716. TPTC2WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD0																															
R/W-0h																															

Table 4-751. TPTC2WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC2

4.8.5.8 TPTC2WRMPUENDADD1 Register (Offset = 124h) [reset = 0h]

TPTC2WRMPUENDADD1 is shown in Figure 4-1003 and described in Table 4-1048.

Return to Summary Table.

Figure 4-717. TPTC2WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD1																															
R/W-0h																															

Table 4-752. TPTC2WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC2

4.8.5.9 TPTC2WRMPUENDADD2 Register (Offset = 128h) [reset = 0h]

TPTC2WRMPUENDADD2 is shown in Figure 4-1004 and described in Table 4-1049.

Return to Summary Table.

Figure 4-718. TPTC2WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD2																															
R/W-0h																															

Table 4-753. TPTC2WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC2

4.8.5.10 TPTC2WRMPUENDADD3 Register (Offset = 12Ch) [reset = 0h]

TPTC2WRMPUENDADD3 is shown in Figure 4-1005 and described in Table 4-1050.

Return to Summary Table.

Figure 4-719. TPTC2WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD3																															
R/W-0h																															

Table 4-754. TPTC2WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC2

4.8.5.11 TPTC2WRMPUENDADD4 Register (Offset = 130h) [reset = 0h]

TPTC2WRMPUENDADD4 is shown in Figure 4-1006 and described in Table 4-1051.

Return to Summary Table.

Figure 4-720. TPTC2WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD4																															
R/W-0h																															

Table 4-755. TPTC2WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC2

4.8.5.12 TPTC2WRMPUENDADD5 Register (Offset = 134h) [reset = 0h]

TPTC2WRMPUENDADD5 is shown in Figure 4-1007 and described in Table 4-1052.

Return to Summary Table.

Figure 4-721. TPTC2WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD5																															
R/W-0h																															

Table 4-756. TPTC2WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC2

4.8.5.13 TPTC2WRMPUERRADD Register (Offset = 140h) [reset = 0h]

TPTC2WRMPUERRADD is shown in Figure 4-1008 and described in Table 4-1053.

Return to Summary Table.

Figure 4-722. TPTC2WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUERRADD																															
R-0h																															

Table 4-757. TPTC2WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC2

4.8.5.14 TPTC2RDMPUSTADD0 Register (Offset = 148h) [reset = 0h]

TPTC2RDMPUSTADD0 is shown in Figure 4-1009 and described in Table 4-1054.

Return to Summary Table.

Figure 4-723. TPTC2RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD0																															
R/W-0h																															

Table 4-758. TPTC2RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC2

4.8.5.15 TPTC2RDMPUSTADD1 Register (Offset = 14Ch) [reset = 0h]

TPTC2RDMPUSTADD1 is shown in Figure 4-1010 and described in Table 4-1055.

Return to Summary Table.

Figure 4-724. TPTC2RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD1																															
R/W-0h																															

Table 4-759. TPTC2RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC2

4.8.5.16 TPTC2RDMPUSTADD2 Register (Offset = 150h) [reset = 0h]

TPTC2RDMPUSTADD2 is shown in Figure 4-1011 and described in Table 4-1056.

Return to Summary Table.

Figure 4-725. TPTC2RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD2																															
R/W-0h																															

Table 4-760. TPTC2RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC2

4.8.5.17 TPTC2RDMPUSTADD3 Register (Offset = 154h) [reset = 0h]

TPTC2RDMPUSTADD3 is shown in Figure 4-1012 and described in Table 4-1057.

Return to Summary Table.

Figure 4-726. TPTC2RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD3																															
R/W-0h																															

Table 4-761. TPTC2RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC2

4.8.5.18 TPTC2RDMPUSTADD4 Register (Offset = 158h) [reset = 0h]

TPTC2RDMPUSTADD4 is shown in Figure 4-1013 and described in Table 4-1058.

Return to Summary Table.

Figure 4-727. TPTC2RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD4																															
R/W-0h																															

Table 4-762. TPTC2RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC2

4.8.5.19 TPTC2RDMPUSTADD5 Register (Offset = 15Ch) [reset = 0h]

TPTC2RDMPUSTADD5 is shown in Figure 4-1014 and described in Table 4-1059.

Return to Summary Table.

Figure 4-728. TPTC2RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD5																															
R/W-0h																															

Table 4-763. TPTC2RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC2

4.8.5.20 TPTC2RDMPUENDADD0 Register (Offset = 168h) [reset = 0h]

TPTC2RDMPUENDADD0 is shown in Figure 4-1015 and described in Table 4-1060.

Return to Summary Table.

Figure 4-729. TPTC2RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD0																															
R/W-0h																															

Table 4-764. TPTC2RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC2

4.8.5.21 TPTC2RDMPUENDADD1 Register (Offset = 16Ch) [reset = 0h]

TPTC2RDMPUENDADD1 is shown in Figure 4-1016 and described in Table 4-1061.

Return to Summary Table.

Figure 4-730. TPTC2RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD1																															
R/W-0h																															

Table 4-765. TPTC2RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC2

4.8.5.22 TPTC2RDMPUENDADD2 Register (Offset = 170h) [reset = 0h]

TPTC2RDMPUENDADD2 is shown in Figure 4-1017 and described in Table 4-1062.

Return to Summary Table.

Figure 4-731. TPTC2RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD2																															
R/W-0h																															

Table 4-766. TPTC2RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC2

4.8.5.23 TPTC2RDMPUENDADD3 Register (Offset = 174h) [reset = 0h]

TPTC2RDMPUENDADD3 is shown in Figure 4-1018 and described in Table 4-1063.

Return to Summary Table.

Figure 4-732. TPTC2RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD3																															
R/W-0h																															

Table 4-767. TPTC2RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC2

4.8.5.24 TPTC2RDMPUENDADD4 Register (Offset = 178h) [reset = 0h]

TPTC2RDMPUENDADD4 is shown in Figure 4-1019 and described in Table 4-1064.

Return to Summary Table.

Figure 4-733. TPTC2RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD4																															
R/W-0h																															

Table 4-768. TPTC2RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC2

4.8.5.25 TPTC2RDMPUENDADD5 Register (Offset = 17Ch) [reset = 0h]

TPTC2RDMPUENDADD5 is shown in Figure 4-1020 and described in Table 4-1065.

Return to Summary Table.

Figure 4-734. TPTC2RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD5																															
R/W-0h																															

Table 4-769. TPTC2RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC2

4.8.5.26 TPTC2RDMPUERRADD Register (Offset = 188h) [reset = 0h]

TPTC2RDMPUERRADD is shown in Figure 4-1021 and described in Table 4-1066.

Return to Summary Table.

Figure 4-735. TPTC2RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUERRADD																															
R-0h																															

Table 4-770. TPTC2RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC2

4.8.5.27 TPTC3WRMPUSTADD0 Register (Offset = 18Ch) [reset = 0h]

TPTC3WRMPUSTADD0 is shown in Figure 4-1022 and described in Table 4-1067.

Return to Summary Table.

Figure 4-736. TPTC3WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD0																															
R/W-0h																															

Table 4-771. TPTC3WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC3

4.8.5.28 TPTC3WRMPUSTADD1 Register (Offset = 190h) [reset = 0h]

TPTC3WRMPUSTADD1 is shown in Figure 4-1023 and described in Table 4-1068.

Return to Summary Table.

Figure 4-737. TPTC3WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD1																															
R/W-0h																															

Table 4-772. TPTC3WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC3

4.8.5.29 TPTC3WRMPUSTADD2 Register (Offset = 194h) [reset = 0h]

TPTC3WRMPUSTADD2 is shown in Figure 4-1024 and described in Table 4-1069.

Return to Summary Table.

Figure 4-738. TPTC3WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD2																															
R/W-0h																															

Table 4-773. TPTC3WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC3

4.8.5.30 TPTC3WRMPUSTADD3 Register (Offset = 198h) [reset = 0h]

TPTC3WRMPUSTADD3 is shown in Figure 4-1025 and described in Table 4-1070.

Return to Summary Table.

Figure 4-739. TPTC3WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD3																															
R/W-0h																															

Table 4-774. TPTC3WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC3

4.8.5.31 TPTC3WRMPUSTADD4 Register (Offset = 19Ch) [reset = 0h]

TPTC3WRMPUSTADD4 is shown in Figure 4-1026 and described in Table 4-1071.

Return to Summary Table.

Figure 4-740. TPTC3WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD4																															
R/W-0h																															

Table 4-775. TPTC3WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC3

4.8.5.32 TPTC3WRMPUSTADD5 Register (Offset = 1A0h) [reset = 0h]

TPTC3WRMPUSTADD5 is shown in Figure 4-1027 and described in Table 4-1072.

Return to Summary Table.

Figure 4-741. TPTC3WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD5																															
R/W-0h																															

Table 4-776. TPTC3WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC3

4.8.5.33 TPTC3WRMPUENDADD0 Register (Offset = 1ACh) [reset = 0h]

TPTC3WRMPUENDADD0 is shown in Figure 4-1028 and described in Table 4-1073.

Return to Summary Table.

Figure 4-742. TPTC3WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD0																															
R/W-0h																															

Table 4-777. TPTC3WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC3

4.8.5.34 TPTC3WRMPUENDADD1 Register (Offset = 1B0h) [reset = 0h]

TPTC3WRMPUENDADD1 is shown in Figure 4-1029 and described in Table 4-1074.

Return to Summary Table.

Figure 4-743. TPTC3WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD1																															
R/W-0h																															

Table 4-778. TPTC3WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC3

4.8.5.35 TPTC3WRMPUENDADD2 Register (Offset = 1B4h) [reset = 0h]

TPTC3WRMPUENDADD2 is shown in Figure 4-1030 and described in Table 4-1075.

Return to Summary Table.

Figure 4-744. TPTC3WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD2																															
R/W-0h																															

Table 4-779. TPTC3WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC3

4.8.5.36 TPTC3WRMPUENDADD3 Register (Offset = 1B8h) [reset = 0h]

TPTC3WRMPUENDADD3 is shown in Figure 4-1031 and described in Table 4-1076.

Return to Summary Table.

Figure 4-745. TPTC3WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD3																															
R/W-0h																															

Table 4-780. TPTC3WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC3

4.8.5.37 TPTC3WRMPUENDADD4 Register (Offset = 1BCh) [reset = 0h]

TPTC3WRMPUENDADD4 is shown in Figure 4-1032 and described in Table 4-1077.

Return to Summary Table.

Figure 4-746. TPTC3WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD4																															
R/W-0h																															

Table 4-781. TPTC3WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC3

4.8.5.38 TPTC3WRMPUENDADD5 Register (Offset = 1C0h) [reset = 0h]

TPTC3WRMPUENDADD5 is shown in Figure 4-1033 and described in Table 4-1078.

Return to Summary Table.

Figure 4-747. TPTC3WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD5																															
R/W-0h																															

Table 4-782. TPTC3WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC3

4.8.5.39 TPTC3WRMPUERRADD Register (Offset = 1CCh) [reset = 0h]

TPTC3WRMPUERRADD is shown in Figure 4-1034 and described in Table 4-1079.

Return to Summary Table.

Figure 4-748. TPTC3WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUERRADD																															
R-0h																															

Table 4-783. TPTC3WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC3

4.8.5.40 TPTC3RDMPUSTADD0 Register (Offset = 1D0h) [reset = 0h]

TPTC3RDMPUSTADD0 is shown in Figure 4-1035 and described in Table 4-1080.

Return to Summary Table.

Figure 4-749. TPTC3RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD0																															
R/W-0h																															

Table 4-784. TPTC3RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC3

4.8.5.41 TPTC3RDMPUSTADD1 Register (Offset = 1D4h) [reset = 0h]

TPTC3RDMPUSTADD1 is shown in Figure 4-1036 and described in Table 4-1081.

Return to Summary Table.

Figure 4-750. TPTC3RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD1																															
R/W-0h																															

Table 4-785. TPTC3RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC3

4.8.5.42 TPTC3RDMPUSTADD2 Register (Offset = 1D8h) [reset = 0h]

TPTC3RDMPUSTADD2 is shown in Figure 4-1037 and described in Table 4-1082.

Return to Summary Table.

Figure 4-751. TPTC3RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD2																															
R/W-0h																															

Table 4-786. TPTC3RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC3

4.8.5.43 TPTC3RDMPUSTADD3 Register (Offset = 1DCh) [reset = 0h]

TPTC3RDMPUSTADD3 is shown in Figure 4-1038 and described in Table 4-1083.

Return to Summary Table.

Figure 4-752. TPTC3RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD3																															
R/W-0h																															

Table 4-787. TPTC3RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC3

4.8.5.44 TPTC3RDMPUSTADD4 Register (Offset = 1E0h) [reset = 0h]

TPTC3RDMPUSTADD4 is shown in Figure 4-1039 and described in Table 4-1084.

Return to Summary Table.

Figure 4-753. TPTC3RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD4																															
R/W-0h																															

Table 4-788. TPTC3RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC3

4.8.5.45 TPTC3RDMPUSTADD5 Register (Offset = 1E4h) [reset = 0h]

TPTC3RDMPUSTADD5 is shown in Figure 4-1040 and described in Table 4-1085.

Return to Summary Table.

Figure 4-754. TPTC3RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD5																															
R/W-0h																															

Table 4-789. TPTC3RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC3

4.8.5.46 TPTC3RDMPUENDADD0 Register (Offset = 1F0h) [reset = 0h]

TPTC3RDMPUENDADD0 is shown in Figure 4-1041 and described in Table 4-1086.

Return to Summary Table.

Figure 4-755. TPTC3RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD0																															
R/W-0h																															

Table 4-790. TPTC3RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC3

4.8.5.47 TPTC3RDMPUENDADD1 Register (Offset = 1F4h) [reset = 0h]

TPTC3RDMPUENDADD1 is shown in Figure 4-1042 and described in Table 4-1087.

Return to Summary Table.

Figure 4-756. TPTC3RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD1																															
R/W-0h																															

Table 4-791. TPTC3RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC3

4.8.5.48 TPTC3RDMPUENDADD2 Register (Offset = 1F8h) [reset = 0h]

TPTC3RDMPUENDADD2 is shown in Figure 4-1043 and described in Table 4-1088.

Return to Summary Table.

Figure 4-757. TPTC3RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD2																															
R/W-0h																															

Table 4-792. TPTC3RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC3

4.8.5.49 TPTC3RDMPUENDADD3 Register (Offset = 1FCh) [reset = 0h]

TPTC3RDMPUENDADD3 is shown in Figure 4-1044 and described in Table 4-1089.

Return to Summary Table.

Figure 4-758. TPTC3RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD3																															
R/W-0h																															

Table 4-793. TPTC3RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC3

4.8.5.50 TPTC3RDMPUENDADD4 Register (Offset = 200h) [reset = 0h]

TPTC3RDMPUENDADD4 is shown in Figure 4-1045 and described in Table 4-1090.

Return to Summary Table.

Figure 4-759. TPTC3RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD4																															
R/W-0h																															

Table 4-794. TPTC3RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC3

4.8.5.51 TPTC3RDMPUENDADD5 Register (Offset = 204h) [reset = 0h]

TPTC3RDMPUENDADD5 is shown in Figure 4-1046 and described in Table 4-1091.

Return to Summary Table.

Figure 4-760. TPTC3RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD5																															
R/W-0h																															

Table 4-795. TPTC3RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC3

4.8.5.52 TPTC3RDMPUERRADD Register (Offset = 210h) [reset = 0h]

TPTC3RDMPUERRADD is shown in Figure 4-1047 and described in Table 4-1092.

Return to Summary Table.

Figure 4-761. TPTC3RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUERRADD																															
R-0h																															

Table 4-796. TPTC3RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC3

4.8.5.53 TPTCMPUVALIDCFG2 Register (Offset = 214h) [reset = 0h]

TPTCMPUVALIDCFG2 is shown in Figure 4-1048 and described in Table 4-1093.

Return to Summary Table.

Figure 4-762. TPTCMPUVALIDCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPTC3RDMPURNGVLD								TPTC3WRMPURNGVLD							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPURNGVLD								TPTC2WRMPURNGVLD							
R/W-0h								R/W-0h							

Table 4-797. TPTCMPUVALIDCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TPTC3RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC3. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
23-16	TPTC3WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC3. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
15-8	TPTC2RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC2. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
7-0	TPTC2WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC2. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled

4.8.5.54 TPTCMPUENCFG2 Register (Offset = 218h) [reset = 0h]

TPTCMPUENCFG2 is shown in Figure 4-1049 and described in Table 4-1094.

Return to Summary Table.

Figure 4-763. TPTCMPUENCFG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPTC3RDMPUERRCLR	TPTC3WRMPUERRCLR	TPTC2RDMPUERRCLR	TPTC2WRMPUERRCLR	TPTC3RDMPUEN	TPTC3WRMPUEN	TPTC2RDMPUEN	TPTC2WRMPUEN
0h	0h	0h	0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-798. TPTCMPUENCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPTC3RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC3. Write 0x1 to clear the MPU error
6	TPTC3WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC3. Write 0x1 to clear the MPU error
5	TPTC2RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC2. Write 0x1 to clear the MPU error
4	TPTC2WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC2. Write 0x1 to clear the MPU error
3	TPTC3RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC3. 0 : MPU is disabled 1 : MPU is enabled
2	TPTC3WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC3. 0 : MPU is disabled 1 : MPU is enabled
1	TPTC2RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC2. 0 : MPU is disabled 1 : MPU is enabled
0	TPTC2WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC2. 0 : MPU is disabled 1 : MPU is enabled

4.8.5.55 L3ECCCFG1 Register (Offset = 268h) [reset = 0h]

L3ECCCFG1 is shown in Figure 4-1050 and described in Table 4-1095.

Return to Summary Table.

Figure 4-764. L3ECCCFG1 Register

31	30	29	28	27	26	25	24
NU					L3ECCREPAIREDBIT		
R-0h					R-0h		
23	22	21	20	19	18	17	16
L3ECCREPAIREDBIT							
R-0h							
15	14	13	12	11	10	9	8
L3ECCREPAIREDBIT							
R-0h							
7	6	5	4	3	2	1	0
L3ECCREPAIREDBIT					L3ECCERRSTAT	L3ECCERRCLR	L3ECCEN
R-0h					R-0h	0h	R/W-0h

Table 4-799. L3ECCCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU	R	0h	Not used
26-3	L3ECCREPAIREDBIT	R	0h	Bit position of repaired bit in L3 ECC memory. Each 6 bits out is this register maps to the corresponding 32 bit location in the data.
2	L3ECCERRSTAT	R	0h	Latched status for L3 ECC error.
1	L3ECCERRCLR		0h	Clear bit for L3 ECC.
0	L3ECCEN	R/W	0h	Enable for L3 ECC logic

4.8.5.56 L3ECCCFG2 Register (Offset = 26Ch) [reset = 0h]

L3ECCCFG2 is shown in Figure 4-1051 and described in Table 4-1096.

Return to Summary Table.

Figure 4-765. L3ECCCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU															L3ECCFAULTADDR																
R-0h															R-0h																

Table 4-800. L3ECCCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU	R	0h	Not used
16-0	L3ECCFAULTADDR	R	0h	Fault address of L3 ECC memory.

4.8.5.57 DSS2MSSSWIRQ Register (Offset = 270h) [reset = 0h]

DSS2MSSSWIRQ is shown in Figure 4-1052 and described in Table 4-1097.

Return to Summary Table.

Figure 4-766. DSS2MSSSWIRQ Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU						MSSSWIRQ2	MSSSWIRQ1
R-0h						0h	0h

Table 4-801. DSS2MSSSWIRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	0h	Not used
1	MSSSWIRQ2		0h	single bit and self clearing interrupt. writing a '1' to this bit will generate a pulse from DSS to MSS VIM line 61
0	MSSSWIRQ1		0h	single bit and self clearing interrupt. writing a '1' to this bit will generate a pulse from DSS to MSS VIM line 52

4.9 18xx Control Registers

4.9.1 MSS_TOPRCM Registers

Table 4-802 lists the memory-mapped registers for the MSS_TOPRCM. All register offset addresses not listed in Table 4-802 should be considered as reserved locations and the register contents should not be modified.

Table 4-802. MSS_TOPRCM Registers

Offset	Acronym	Register Name	Section
8h	BSSCTL	Control Signals to BSS	Section 4.9.1.1
Ch	DSSCTL	Control Signals to DSS	Section 4.9.1.2
10h	EXTCLKDIV	Clock divide value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.3
14h	EXTCLKSRCSEL	Clock source select value for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.4
18h	EXTCLKCTL	Clock gate control for MCU_CLKOUT and PMIC_CLKOUT	Section 4.9.1.5
1Ch	SOFTSYSRST	Software triggered Warm Reset	Section 4.9.1.6
20h	WDRSTEN	Issue Warm reset upon MSS Watch dog reset	Section 4.9.1.7
24h	SYSRSTCAUSE	Reset cause register	Section 4.9.1.8
28h	SYSRSTCAUSECLR	Clear Reset Cause register	Section 4.9.1.9
34h	MISCCAPT	Capture required Status values across the chip.	Section 4.9.1.10
38h	DCDCCTL0	PMIC_CLKOUT dethering control	Section 4.9.1.11
3Ch	DCDCCTL1	PMIC_CLKOUT dethering control	Section 4.9.1.12
48h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.1.13
4Ch	LVDSPADCTL0	LVDS pad control	Section 4.9.1.14
50h	LVDSPADCTL1	LVDS pad control	Section 4.9.1.15
60h	DFTREG0		Section 4.9.1.16
64h	DFTREG1		Section 4.9.1.17
74h	DFTREG5		Section 4.9.1.18
DCh	MEMINITDONE	Memory Initialization done status for memories in BSS and DSS	Section 4.9.1.19
ECh	SPARE0_3	Spare registers 0 to 3	Section 4.9.1.20
FCh	MSS_SIGNATURE	MSS Signature register	Section 4.9.1.21
158h	GEMBOOTSTCEN		Section 4.9.1.22
178h	MISCCTL1	Miscellaneous Control Register	Section 4.9.1.23
180h	USERMODEEN2	USER_MODE_ACCESS_EN	Section 4.9.1.24
18Ch	SYSTICK		Section 4.9.1.25
1C4h	SECURECFGREG1		Section 4.9.1.26
1C8h	SECURECFGREG2		Section 4.9.1.27
1CCh	SECURECFGREG3		Section 4.9.1.28
1D0h	SECURECFGREG4		Section 4.9.1.29
1D4h	SECURERAMREG		Section 4.9.1.30
1E4h	SPAREMULTIBIT	MIBSPI setup register	Section 4.9.1.31
1E8h	SPARE4_9	Spare registers 4 to 9	Section 4.9.1.32
200h	UID31TO0	Efuse Row Capture register for FROM1	Section 4.9.1.33
204h	UID63TO32	Efuse Row Capture register for FROM1	Section 4.9.1.34
208h	UID95TO64	Efuse Row Capture register for FROM1	Section 4.9.1.35
20Ch	UID119TO96	Efuse Row Capture register for FROM1	Section 4.9.1.36
2A8h	MEMINITSTARTSHMEM	Shared memory initialization start	Section 4.9.1.37
2ACh	MEMINITDONESHMEM	Shared memory intialization end	Section 4.9.1.38
2B0h	DSSMEMTAB0	Controls ordering of banks in shared memory associated with DSS	Section 4.9.1.39
2BCh	TCMAMEMTAB	Controls ordering of banks in shared memory associated with MSS TCMA	Section 4.9.1.40
2C0h	TCMBMEMTAB	Controls ordering of banks in shared memory associated with MSS TCMB	Section 4.9.1.41
2C8h	SHMEMBANKSEL3TO0	Shared memory master allocation.	Section 4.9.1.42

Table 4-802. MSS_TOPRCM Registers (continued)

Offset	Acronym	Register Name	Section
2CCh	SHMEMBANKSEL7TO4	Shared memory master allocation.	Section 4.9.1.43
2D0h	PBISTCLKCTL	PBIST clock control register	Section 4.9.1.44

Complex bit access types are encoded to fit into small table cells. Table 4-803 shows the codes that are used for access types in this section.

Table 4-803. MSS_TOPRCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.9.1.1 BSSCTL Register (Offset = 8h) [reset = ADADADADh]

BSSCTL is shown in Figure 4-767 and described in Table 4-804.

Return to Summary Table.

Control Signals to BSS

Figure 4-767. BSSCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BSSCPUHALT								RESERVED							
R/W-ADh								R/W-ADh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
R/W-ADh															

Table 4-804. BSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BSSCPUHALT	R/W	ADh	Halt BSS CR4, To halt : either 3:0 should be 0xD or 7:4 should be 0xA One should Halt the processor before releasing BSS reset.
23-0	RESERVED	R/W	ADh	Reserved

4.9.1.2 DSSCTL Register (Offset = Ch) [reset = 00ADADADh]

DSSCTL is shown in Figure 4-768 and described in Table 4-805.

Return to Summary Table.

Control Signals to DSS

Figure 4-768. DSSCTL Register

31	30	29	28	27	26	25	24
NU					GEMLRSTN	GEMGRSTN	GEMPORZ
0h					R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-ADh							
15	14	13	12	11	10	9	8
RESERVED							
R/W-ADh							
7	6	5	4	3	2	1	0
RESERVED							
R/W-ADh							

Table 4-805. DSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU		0h	Reserved
26	GEMLRSTN	R/W	0h	DSP Local Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
25	GEMGRSTN	R/W	0h	DSP Global Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
24	GEMPORZ	R/W	0h	DSP Por Reset value that will be propagated to the DSP when the DSP Power FSM has switched the DSP on. 0x0 : Reset is asserted 0x1 : Reset is deasserted This value is to be set by SW to 0x1 once initially before the DSP is powered on and is not expected to be changed. All Reset toggling for the DSP is exepcted to be done via the DSP Power FSM and the DSP STC FSM
23-0	RESERVED	R/W	ADh	Reserved

4.9.1.3 EXTCLKDIV Register (Offset = 10h) [reset = 0h]

EXTCLKDIV is shown in Figure 4-769 and described in Table 4-806.

Return to Summary Table.

Clock divide value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-769. EXTCLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2DIV								EXTCLK1DIV							
R/W-0h								R/W-0h							

Table 4-806. EXTCLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Reserved
15-8	EXTCLK2DIV	R/W	0h	Divide value for clock source (selected by field EXTCLKSRCSEL in register CLKSRCSEL.) to PMIC_CLKOUT generation logic “0000_0000” => div1 “0000_0001” => div2 | | “1111_1111” => div256 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming EXTCLK2SRCSEL.
7-0	EXTCLK1DIV	R/W	0h	Divide value for MCU_CLKOUT (the one going out of chip) source clock selected by field EXTCLKSRCSEL in register CLKSRCSEL. “0000_0000” => div1 “0000_0001” => div2 | | “1111_1111” => div256 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming EXTCLK1SRCSEL.

4.9.1.4 EXTCLKSRCSEL Register (Offset = 14h) [reset = 0h]

EXTCLKSRCSEL is shown in Figure 4-770 and described in Table 4-807.

Return to Summary Table.

Clock source select value for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-770. EXTCLKSRCSEL Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1				EXTCLK2SRCSEL			
0h				R/W-0h			
7	6	5	4	3	2	1	0
NU0				EXTCLK1SRCSEL			
0h				R/W-0h			

Table 4-807. EXTCLKSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU1		0h	Reserved
11-8	EXTCLK2SRCSEL	R/W	0h	Select clock source for PMIC_CLKOUT generation “000” => CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) “001” => RCCLK (10Mhz) “010” => 600Mhz PLL divided clock “011” => 240Mhz PLL divided clock “100” => RCCLK (10Mhz) “101” => RCCLK (10Mhz) “110” => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mhz) “111” => RCCLK
7-4	NU0		0h	Reserved
3-0	EXTCLK1SRCSEL	R/W	0h	Select clock source for MCU_CLKOUT “000” =>CPUCLK from ANA(XTAL 40Mhz or 50 Mhz or 80Mh or 100Mhz/ RCCLK in WU limp mode) “001” => RCCLK (10Mhz) “010” => 600Mhz PLL divided clock “011” => 240Mhz PLL divided clock “100” => RCCLK (10Mhz) “101” => RCCLK (10Mhz) “110” => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mhz) “111” => RCCLK

4.9.1.5 EXTCLKCTL Register (Offset = 18h) [reset = ADADh]

EXTCLKCTL is shown in Figure 4-771 and described in Table 4-808.

Return to Summary Table.

Clock gate control for MCU_CLKOUT and PMIC_CLKOUT

Figure 4-771. EXTCLKCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTCLK2GATE								EXTCLK1GATE							
R/W-ADh								R/W-ADh							

Table 4-808. EXTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Reserved
15-8	EXTCLK2GATE	R/W	ADh	Pre clock divider gate for PMIC_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA
7-0	EXTCLK1GATE	R/W	ADh	Pre clock divider gate for MCU_CLKOUT. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA

4.9.1.6 SOFTSYSRST Register (Offset = 1Ch) [reset = 0h]

SOFTSYSRST is shown in Figure 4-772 and described in Table 4-809.

Return to Summary Table.

Software triggered Warm Reset

Figure 4-772. SOFTSYSRST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SOFTSYSRST							
0h								R/W-0h							

Table 4-809. SOFTSYSRST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	SOFTSYSRST	R/W	0h	Write 0xAD to trigger warm reset to the chip.

4.9.1.7 WDRSTEN Register (Offset = 20h) [reset = 0h]

WDRSTEN is shown in Figure 4-773 and described in Table 4-810.

Return to Summary Table.

Issue Warm reset upon MSS Watch dog reset

Figure 4-773. WDRSTEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								WDRSTEN							
0h																								R/W-0h							

Table 4-810. WDRSTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	WDRSTEN	R/W	0h	Write 0xAD to trigger warm reset to the chip upon MSS wdog reset.

4.9.1.8 SYSRSTCAUSE Register (Offset = 24h) [reset = 0h]

SYSRSTCAUSE is shown in Figure 4-774 and described in Table 4-811.

Return to Summary Table.

Reset cause register

Figure 4-774. SYSRSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU												SYSRSTCAUSE			
0h												R-0h			

Table 4-811. SYSRSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU		0h	Reserved
3-0	SYSRSTCAUSE	R	0h	Gives cause of chip reset “1001” : System out of NRESET “1010” : Warm reset because of MSS Wdog. “1100” : Warm reset because of Software trigger- SOFTSYSRST “1000” : External Warm Reset Note: The ROM Bootloader clears this register as part of its processing, so it always reads zero by the application or secondary bootloader. However, the ROM Bootloader saves the original value into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.9.1.9 SYSRSTCAUSECLR Register (Offset = 28h) [reset = 0h]

SYSRSTCAUSECLR is shown in Figure 4-775 and described in Table 4-812.

Return to Summary Table.

Clear Reset Cause register

Figure 4-775. SYSRSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								SYSRSTCAUSECLR							
0h								0h							

Table 4-812. SYSRSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	SYSRSTCAUSECLR		0h	Write 0xAD to clear SYSRSTCAUSE..Self clearing

4.9.1.10 MISCCAPT Register (Offset = 34h) [reset = 0h]

MISCCAPT is shown in Figure 4-776 and described in Table 4-813.

Return to Summary Table.

Capture required Status values across the chip.

Figure 4-776. MISCCAPT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCAPT																															
R-0h																															

Table 4-813. MISCCAPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCAPT	R	0h	0: No error Any other non-zero value: Error

4.9.1.11 DCDCCTL0 Register (Offset = 38h) [reset = 0h]

DCDCCTL0 is shown in Figure 4-777 and described in Table 4-814.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-777. DCDCCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCDCCTL0																															
R/W-0h																															

Table 4-814. DCDCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DCDCCTL0	R/W	0h	[26:0] PMIC_CLKOUT Frequency slope value.Unsigned

4.9.1.12 DCDCCTL1 Register (Offset = 3Ch) [reset = 0h]

DCDCCTL1 is shown in Figure 4-778 and described in Table 4-815.

Return to Summary Table.

PMIC_CLKOUT dethering control

Figure 4-778. DCDCCTL1 Register

31	30	29	28	27	26	25	24
DCDCCTL1							
R/W-0h							
23	22	21	20	19	18	17	16
DCDCCTL1							
R/W-0h							
15	14	13	12	11	10	9	8
DCDCCTL1							
R/W-0h							
7	6	5	4	3	2	1	0
DCDCCTL1						DCDCLKEN	DCDCRST
R/W-0h						R/W-0h	R/W-0h

Table 4-815. DCDCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	DCDCCTL1	R/W	0h	PMIC_CLKOUT dithering control [8] : Frequency accumulation Mode . ‘0’ : Continous mode.’1’ : Staircase mode [9] : ‘1’ Enables dither [23:16] : Minimum frequency threshold [31:24] : Maximum frequency threshold
1	DCDCLKEN	R/W	0h	PMIC_CLKOUT Enable – Multi Bit
0	DCDCRST	R/W	0h	PMIC_CLKOUT dithering control block reset (active high) – Multi Bit

4.9.1.13 USERMODEEN Register (Offset = 48h) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-779. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-816. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space from 0x00 to 0XFF

4.9.1.14 LVDSPADCTL0 Register (Offset = 4Ch) [reset = 01010101h]

LVDSPADCTL0 is shown in Figure 4-780 and described in Table 4-817.

Return to Summary Table.

LVDS pad control

Figure 4-780. LVDSPADCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL0																															
R/W-01010101h																															

Table 4-817. LVDSPADCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL0	R/W	01010101h	0 : pwrdn Control for i_LVDSclk_io_cell 1 : lopwra Control for i_LVDSclk_io_cell 2 : lopwrb Control for i_LVDSclk_io_cell 3 : lpsel Control for i_LVDSclk_io_cell 4 : sub_lvds_en Control for i_LVDSclk_io_cell 5 : hiz_disable Control for i_LVDSclk_io_cell 6 : ext_res_en Control for i_LVDSclk_io_cell 7 : Reserved 8 : pwrdn Control for i_LVDS_tx0_io_cell 9 : lopwra Control for i_LVDS_tx0_io_cell 10 : lopwrb Control for i_LVDS_tx0_io_cell 11 : lpsel Control for i_LVDS_tx0_io_cell 12 : sub_lvds_en Control for i_LVDS_tx0_io_cell 13 : hiz_disable Control for i_LVDS_tx0_io_cell 14 : ext_res_en Control for i_LVDS_tx0_io_cell 15 : Reserved 16 : pwrdn Control for i_LVDS_tx1_io_cell 17 : lopwra Control for i_LVDS_tx1_io_cell 18 : lopwrb Control for i_LVDS_tx1_io_cell 19 : lpsel Control for i_LVDS_tx1_io_cell 20 : sub_lvds_en Control for i_LVDS_tx1_io_cell 21 : hiz_disable Control for i_LVDS_tx1_io_cell 22 : ext_res_en Control for i_LVDS_tx1_io_cell 23 -31 : Reserved

4.9.1.15 LVDSPADCTL1 Register (Offset = 50h) [reset = 101h]

LVDSPADCTL1 is shown in Figure 4-781 and described in Table 4-818.

Return to Summary Table.

LVDS pad control

Figure 4-781. LVDSPADCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LVDSPADCTL1																															
R/W-101h																															

Table 4-818. LVDSPADCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LVDSPADCTL1	R/W	101h	0- 7 : Reserved 8 : pwrdn Control for i_LVDSfrclk_io_cell 9 : lopwra Control for i_LVDSfrclk_io_cell 10 : lopwrb Control for i_LVDSfrclk_io_cell 11 : lpsel Control for i_LVDSfrclk_io_cell 12 : sub_lvds_en Control for i_LVDSfrclk_io_cell 13 : hiz_disable Control for i_LVDSfrclk_io_cell 14 : ext_res_en Control for i_LVDSfrclk_io_cell 15-23: Reserved 24 : pwrdn Control for lvds_bias_cell 25 : efuse_set Control for lvds_bias_cell

4.9.1.16 DFTREG0 Register (Offset = 60h) [reset = 0h]

DFTREG0 is shown in Figure 4-782 and described in Table 4-819.

Return to Summary Table.

Figure 4-782. DFTREG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG0																															
R/W-0h																															

Table 4-819. DFTREG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG0	R/W	0h	[3:0] : MSS PBIST SELFTEST KEY = 4'b1010 [4] : Reserved [5] : PBIST IP reset control. 1 = reset, 0 = out of reset. The PBIST_SELFTEST_KEY must be enabled to program this bit. [31:6] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.9.1.17 DFTREG1 Register (Offset = 64h) [reset = 0h]

DFTREG1 is shown in Figure 4-783 and described in Table 4-820.

Return to Summary Table.

Figure 4-783. DFTREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG1																															
R/W-0h																															

Table 4-820. DFTREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG1	R/W	0h	[31:0] : Write 1'b1 to a bit to configure a particular memory group for self-test, and 1'b0 to disable a particular memory group.

4.9.1.18 DFTREG5 Register (Offset = 74h) [reset = 0h]

DFTREG5 is shown in Figure 4-784 and described in Table 4-821.

Return to Summary Table.

Figure 4-784. DFTREG5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFTREG5																															
R/W-0h																															

Table 4-821. DFTREG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DFTREG5	R/W	0h	[0] : Reserved [4:1] DSP PBIST SELFTEST KEY = 4'b1010 [8:5]: Subsystem level memory self-test power clock gate enable controls [31:9] : Reserved

4.9.1.19 MEMINITDONE Register (Offset = DCh) [reset = 0h]

MEMINITDONE is shown in Figure 4-822 and described in Table 4-861.

Return to Summary Table.

Memory Initialization done status for memories in BSS and DSS

Figure 4-785. MEMINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED						BSSVIMMEM	RESERVED
0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED							
0h							

Table 4-822. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED		0h	Reserved
9	BSSVIMMEM	R	0h	Memory Initialization done status for BSS VIM memory
8-0	RESERVED	R	0h	Reserved

4.9.1.20 SPARE0_3 Register (Offset = ECh, F0h, F4h, F8h) [reset = 0h]

SPARE0_3 is shown in Figure 4-786 and described in Table 4-823.

Return to Summary Table.

Spare Register

Figure 4-786. SPARE0_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE0_3																															
R/W-0h																															

Table 4-823. SPARE0_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE0_3	R/W	0h	Four 32-bit registers that may be used by an application for any purpose.

4.9.1.21 MSS_SIGNATURE Register (Offset = FCh) [reset = 0BB5202Fh]

MSS_SIGNATURE is shown in Figure 4-787 and described in Table 4-824.

Return to Summary Table.

Spare Register

Figure 4-787. MSS_SIGNATURE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSS_SIGNATURE																															
R/W-0BB5202Fh																															

Table 4-824. MSS_SIGNATURE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MSS_SIGNATURE	R/W	0BB5202Fh	

4.9.1.22 GEMBOOTSTCEN Register (Offset = 158h) [reset = 0h]

GEMBOOTSTCEN is shown in Figure 4-788 and described in Table 4-825.

Return to Summary Table.

Figure 4-788. GEMBOOTSTCEN Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							GEMBOOTSTCEN
R-0h							R/W-0h

Table 4-825. GEMBOOTSTCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	
0	GEMBOOTSTCEN	R/W	0h	'1' : Enable GEM STC during GEM power UP

4.9.1.23 MISCCTL1 Register (Offset = 178h) [reset = 0h]

MISCCTL1 is shown in Figure 4-789 and described in Table 4-826.

Return to Summary Table.

Miscellaneous Control Register

Figure 4-789. MISCCTL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISCCTL1																															
R/W-0h																															

Table 4-826. MISCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MISCCTL1	R/W	0h	7:0 : Write 0xAD to enable Warm_resetn from external device in addition to internally generated warm reset 16:8 : Write 0xAD to take board level loop back clock for QSPI. 24:16 : Write 0xAD to external clock as QSPI baud clock source – needed for DFT IO char..

4.9.1.24 USERMODEEN2 Register (Offset = 180h) [reset = 0h]

USERMODEEN2 is shown in Figure 4-790 and described in Table 4-827.

Return to Summary Table.

Figure 4-790. USERMODEEN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN2																															
R/W-0h																															

Table 4-827. USERMODEEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN2	R/W	0h	Write 0XADADADAD to enable user mode write access to TOP RCM space which are resettable only by Power on reset. i.e. from offset address 0x100 to 0x1FF

4.9.1.25 SYSTICK Register (Offset = 18Ch) [reset = 0h]

SYSTICK is shown in Figure 4-791 and described in Table 4-828.

Return to Summary Table.

Figure 4-791. SYSTICK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSTICK																															
R-0h																															

Table 4-828. SYSTICK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYSTICK	R	0h	Continuous counter running on 32Khz derived from RC clock.

4.9.1.26 SECURECFGREG1 Register (Offset = 1C4h) [reset = 00700777h]

SECURECFGREG1 is shown in Figure 4-792 and described in Table 4-829.

Return to Summary Table.

Figure 4-792. SECURECFGREG1 Register

31	30	29	28	27	26	25	24
RESERVED	JTAGFIREWALLEN			RESERVED			
0h	R/W-0h			R/W-0h			
23	22	21	20	19	18	17	16
RESERVED	SECURERAMFIREWALLEN			RESERVED	LOGGERFIREWALLEN		
0h	R/W-7h			0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	TRACEFIREWALLEN			RESERVED	CRYPTOFIREWALLEN		
0h	R/W-0h			0h	R/W-7h		
7	6	5	4	3	2	1	0
RESERVED	CUSTCEKFIREWALLEN			RESERVED			
0h	R/W-7h			R/W-0h			

Table 4-829. SECURECFGREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED		0h	
30-28	JTAGFIREWALLEN	R/W	0h	JTAG Firewall.Firewall Disabled for value "111" and enabled for rest
27-23	RESERVED		0h	
22-20	SECURERAMFIREWALLEN	R/W	7h	Set Secure RAM Firewall.Firewall Disabled for value "111" and enabled for rest
19	RESERVED		0h	
18-16	LOGGERFIREWALLEN	R/W	0h	Set Logger Firewall.Firewall Disabled for value "111" and enabled for rest
15	RESERVED		0h	
14-12	TRACEFIREWALLEN	R/W	0h	Set Trace Firewall. Firewall Disabled for value "111" and enabled for rest
11	RESERVED		0h	
10-8	CRYPTOFIREWALLEN	R/W	7h	Set Crypto Firewall. Firewall Disabled for value "111" and enabled for rest
7	RESERVED		0h	
6-4	CUSTCEKFIREWALLEN	R/W	7h	Set CEK1,CEK2 firewall.Firewall Disabled for value "111" and enabled for rest
3-0	RESERVED		0h	

4.9.1.27 SECURECFGREG2 Register (Offset = 1C8h) [reset = 0h]

SECURECFGREG2 is shown in Figure 4-793 and described in Table 4-830.

Return to Summary Table.

Figure 4-793. SECURECFGREG2 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED					CUSTKEYERASE		
0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					DMMFIREWALLEN		
0h					R/W-0h		

Table 4-830. SECURECFGREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED		0h	Reserved
10-8	CUSTKEYERASE	R/W	0h	Erase CEK1 ,CEK2,CPK Keys when value "111" is written
7-3	RESERVED		0h	Reserved
2-0	DMMFIREWALLEN	R/W	0h	DMM Firewall.Firewall Disabled for value "111" and enabled for rest

4.9.1.28 SECURECFGREG3 Register (Offset = 1CCh) [reset = 0h]

SECURECFGREG3 is shown in Figure 4-794 and described in Table 4-831.

Return to Summary Table.

Figure 4-794. SECURECFGREG3 Register

31	30	29	28	27	26	25	24
RESERVED	JTAGSTICKYBIT			RESERVED			
0h	R/W-0h			0h			
23	22	21	20	19	18	17	16
RESERVED	SECURERAMSTICKYBIT			RESERVED	TRACESTICKYBIT		
0h	R/W-0h			0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	CRYPTOSTICKYBIT			RESERVED	CUSTCEKSTICKYBIT		
0h	R/W-0h			0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					LOGGERSTICKYBIT		
0h					R/W-0h		

Table 4-831. SECURECFGREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED		0h	
30-28	JTAGSTICKYBIT	R/W	0h	JTAG Sticky Reg. Sticky reg is set when value "111 is written
27-23	RESERVED		0h	
22-20	SECURERAMSTICKYBIT	R/W	0h	Secure RAM Sticky Reg. Sticky reg is set when value "111 is written
19	RESERVED		0h	
18-16	TRACESTICKYBIT	R/W	0h	Trace Sticky Reg. Sticky reg is set when value "111 is written
15	RESERVED		0h	
14-12	CRYPTOSTICKYBIT	R/W	0h	Crypto Sticky Reg. Sticky reg is set when value "111 is written
11	RESERVED		0h	
10-8	CUSTCEKSTICKYBIT	R/W	0h	CEK1,CEK2 Sticky Reg for firewall. Sticky reg is set when value "111 is written
7-3	RESERVED		0h	
2-0	LOGGERSTICKYBIT	R/W	0h	Logger Sticky Reg. Sticky reg is set when value "111 is written

4.9.1.29 SECURECFGREG4 Register (Offset = 1D0h) [reset = 0h]

SECURECFGREG4 is shown in Figure 4-795 and described in Table 4-832.

Return to Summary Table.

Figure 4-795. SECURECFGREG4 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RESERVED					DMMSTICKYBIT		
0h					R/W-0h		

Table 4-832. SECURECFGREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED		0h	Reserved
2-0	DMMSTICKYBIT	R/W	0h	DMM Sticky Reg. Sticky reg is set when value "111 is written

4.9.1.30 SECURERAMREG Register (Offset = 1D4h) [reset = 0h]

SECURERAMREG is shown in Figure 4-796 and described in Table 4-833.

Return to Summary Table.

Figure 4-796. SECURERAMREG Register

31	30	29	28	27	26	25	24
NU4							SECURERAMKEY255
0h							R/W-0h
23	22	21	20	19	18	17	16
SECURERAMKEYIDX							
R/W-0h							
15	14	13	12	11	10	9	8
NU2							SECURERAMKEYRD
0h							0h
7	6	5	4	3	2	1	0
NU10							SECURERAMRDDONE
0h							R-0h

Table 4-833. SECURERAMREG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4		0h	
24	SECURERAMKEY255	R/W	0h	1: Secure RAM key bitwidth = 255 0:Secure RAM key bitwidth = 128
23-16	SECURERAMKEYIDX	R/W	0h	Index to Secure RAM
15-9	NU2		0h	
8	SECURERAMKEYRD		0h	1: Load key from secure RAM into register to be used by AES engine.Self clearing
7-1	NU10		0h	
0	SECURERAMRDDONE	R	0h	Secure RAM key loaded into register

4.9.1.31 SPAREMULTIBIT Register (Offset = 1E4h) [reset = FFFF0000h]

SPAREMULTIBIT is shown in Figure 4-797 and described in Table 4-834.

Return to Summary Table.

Spare Register

Figure 4-797. SPAREMULTIBIT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-1h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-1h							
15	14	13	12	11	10	9	8
RESERVED				SPAREMULTIBIT11	SPAREMULTIBIT10	SPAREMULTIBIT9	SPAREMULTIBIT8
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SPAREMULTIBIT7	SPAREMULTIBIT6	SPAREMULTIBIT5	SPAREMULTIBIT4	SPAREMULTIBIT3	SPAREMULTIBIT2	SPAREMULTIBIT1	SPAREMULTIBIT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-834. SPAREMULTIBIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	1h	Reserved
11	SPAREMULTIBIT11	R/W	0h	MIBSPIB : when set the TRIGGER's are un-gated only when chip-select is active
10	SPAREMULTIBIT10	R/W	0h	SPIB trigger source polarity select.'0' - Polarity 0,'1'-Polarity 1
9	SPAREMULTIBIT9	R/W	0h	SPIA trigger source polarity select.'0' - Polarity 0,'1'-Polarity 1
8	SPAREMULTIBIT8	R/W	0h	'1': MIBSPIB External chip select is overridden with the value of MIBSPIB CS polarity-slave mode
7	SPAREMULTIBIT7	R/W	0h	'1': MIBSPIA External chip select is overridden with the value of MIBSPIA CS polarity-slave mode
6	SPAREMULTIBIT6	R/W	0h	MIBSPIB CS Trigger SRC enable '1': Use CS as trigger source
5	SPAREMULTIBIT5	R/W	0h	MIBSPIB CS polarity-slave mode 1: Active high 0:Active low
4	SPAREMULTIBIT4	R/W	0h	MIBSPIB MISO OE_N Control based on Chip select(CS)-applicable in slave mode 1:MISO OEN controlled based on CS.When CS is inactive OE_N=1 0:MISO OEN controlled by IP
3	SPAREMULTIBIT3	R/W	0h	MIBSPIA :When set the TRIGGER's are un-gated only when chip-select is active.
2	SPAREMULTIBIT2	R/W	0h	MIBSPIA CS Trigger SRC enable-slave mode '1': Use CS as trigger source
1	SPAREMULTIBIT1	R/W	0h	MIBSPIA CS polarity-slave mode 1: Active high 0:Active low
0	SPAREMULTIBIT0	R/W	0h	MIBSPIA MISO OE_N Control based on Chip select(CS)-applicable in slave mode 1:MISO OEN controlled based on CS.When CS is inactive OE_N=1,else controlled by IP 0:MISO OEN controlled by IP

4.9.1.32 SPARE4_9 Register (Offset = 1E8h, 1ECh, … 1FCh) [reset = 0h]

SPARE4_9 is shown in Figure 4-798 and described in Table 4-835.

Return to Summary Table.

Figure 4-798. SPARE4_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID31TO0																															
R-0h																															

Table 4-835. SPARE4_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE4_9	R	0h	Six 32-bit registers that may be used by an application for any purpose. SPARE4 through SPARE8 are not used by the ROM Bootloader, and will survive a Warm Reset. SPARE9 is used by the ROM Bootloader to record RCM_RSTCAUSE and TOPRCM_SYSRSTCAUSE before they are modified by the Bootloader. SPARE 9 is written by the Bootloader thusly: SPARE9[3:0] - current TOPRCM reset cause SPARE9[11:4] - current MSS RCM reset cause SPARE9[15:12] - UNUSED SPARE9[19:16] - previous TOP RCM reset cause SPARE9[27:20] - previous MSS RCM reset cause SPARE9[31:38] - UNUSED

4.9.1.33 UID31TO0 Register (Offset = 200h) [reset = 0h]

UID31TO0 is shown in Figure 4-799 and described in Table 4-836.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-799. UID31TO0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID31TO0																															
R-0h																															

Table 4-836. UID31TO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID31TO0	R	0h	FROM1 Efuse Read UID[31:0]

4.9.1.34 UID63TO32 Register (Offset = 204h) [reset = 0h]

UID63TO32 is shown in Figure 4-800 and described in Table 4-837.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-800. UID63TO32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID63TO32																															
R-0h																															

Table 4-837. UID63TO32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID63TO32	R	0h	FROM1 Efuse Read UID[63:32]

4.9.1.35 UID95TO64 Register (Offset = 208h) [reset = 0h]

UID95TO64 is shown in Figure 4-801 and described in Table 4-838.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-801. UID95TO64 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID95TO64																															
R-0h																															

Table 4-838. UID95TO64 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID95TO64	R	0h	FROM1 Efuse Read UID[95:64]

4.9.1.36 UID119TO96 Register (Offset = 20Ch) [reset = 0h]

UID119TO96 is shown in Figure 4-802 and described in Table 4-839.

Return to Summary Table.

Efuse Row Capture register for FROM1

Figure 4-802. UID119TO96 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID119TO96																															
R-0h																															

Table 4-839. UID119TO96 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID119TO96	R	0h	FROM1 Efuse Read UID[119:96]

4.9.1.37 MEMINITSTARTSHMEM Register (Offset = 2A8h) [reset = 0h]

MEMINITSTARTSHMEM is shown in Figure 4-803 and described in Table 4-840.

Return to Summary Table.

Shared memory initialization start

Figure 4-803. MEMINITSTARTSHMEM Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
MEMINITSTARTBANK7	MEMINITSTARTBANK6	MEMINITSTARTBANK5	MEMINITSTARTBANK4	MEMINITSTARTBANK3	MEMINITSTARTBANK2	MEMINITSTARTBANK1	MEMINITSTARTBANK0
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-840. MEMINITSTARTSHMEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1		0h	Not used
7	MEMINITSTARTBANK7		0h	Writing '1' will trigger Shared memory initialization for bank 7.Self clearing
6	MEMINITSTARTBANK6		0h	Writing '1' will trigger Shared memory initialization for bank 6.Self clearing
5	MEMINITSTARTBANK5		0h	Writing '1' will trigger Shared memory initialization for bank 5.Self clearing
4	MEMINITSTARTBANK4		0h	Writing '1' will trigger Shared memory initialization for bank 4.Self clearing
3	MEMINITSTARTBANK3		0h	Writing '1' will trigger Shared memory initialization for bank 3.Self clearing
2	MEMINITSTARTBANK2		0h	Writing '1' will trigger Shared memory initialization for bank 2.Self clearing
1	MEMINITSTARTBANK1		0h	Writing '1' will trigger Shared memory initialization for bank 1.Self clearing
0	MEMINITSTARTBANK0		0h	Writing '1' will trigger Shared memory initialization for bank 0.Self clearing

4.9.1.38 MEMINITDONESHMEM Register (Offset = 2ACh) [reset = 0h]

MEMINITDONESHMEM is shown in Figure 4-804 and described in Table 4-841.

Return to Summary Table.

Shared memory intialization end

Figure 4-804. MEMINITDONESHMEM Register

31	30	29	28	27	26	25	24
NU1							
0h							
23	22	21	20	19	18	17	16
NU1							
0h							
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
MEMINITDONEBANK7	MEMINITDONEBANK6	MEMINITDONEBANK5	MEMINITDONEBANK4	MEMINITDONEBANK3	MEMINITDONEBANK2	MEMINITDONEBANK1	MEMINITDONEBANK0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 4-841. MEMINITDONESHMEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1		0h	Not used
7	MEMINITDONEBANK7	R	0h	Memory Initialization done status for Shared memory for bank 7
6	MEMINITDONEBANK6	R	0h	Memory Initialization done status for Shared memory for bank 6
5	MEMINITDONEBANK5	R	0h	Memory Initialization done status for Shared memory for bank 5
4	MEMINITDONEBANK4	R	0h	Memory Initialization done status for Shared memory for bank 4
3	MEMINITDONEBANK3	R	0h	Memory Initialization done status for Shared memory for bank 3
2	MEMINITDONEBANK2	R	0h	Memory Initialization done status for Shared memory for bank 2
1	MEMINITDONEBANK1	R	0h	Memory Initialization done status for Shared memory for bank 1
0	MEMINITDONEBANK0	R	0h	Memory Initialization done status for Shared memory for bank 0

4.9.1.39 DSSMEMTAB0 Register (Offset = 2B0h) [reset = 76543210h]

DSSMEMTAB0 is shown in Figure 4-805 and described in Table 4-842.

Return to Summary Table.

Controls ordering of banks in shared memory associated with DSS

Figure 4-805. DSSMEMTAB0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSSMEMTAB0																															
R/W-76543210h																															

Table 4-842. DSSMEMTAB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DSSMEMTAB0	R/W	76543210h	DSS L3RAM memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 0), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 1), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 2), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 3). 4th 128KB address goes to bank number programmed in [19:16] of this register(default is bank 4), 5th 128KB address goes to bank number programmed in [23:20] of this register(default is bank 5), 6th 128KB address goes to bank number programmed in [27:24] of this register(default is bank 6), 7th 128KB address goes to bank number programmed in [31:28] of this register(default is bank 7). Corresponding banks need to be selected for DSS in SHMEMBANKSEL regsiter

4.9.1.40 TCMAMEMTAB Register (Offset = 2BCh) [reset = 7654h]

TCMAMEMTAB is shown in Figure 4-806 and described in Table 4-843.

Return to Summary Table.

Controls ordering of banks in shared memory associated with MSS TCMA

Figure 4-806. TCMAMEMTAB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																TCMAMEMTAB															
0h																R/W-7654h															

Table 4-843. TCMAMEMTAB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU1		0h	Not Used
15-0	TCMAMEMTAB	R/W	7654h	MSS TCMA memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 4), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 5), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 6), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 7). Corresponding banks need to be selected for MSS TCMA in SHMEMBANKSEL regsiter

4.9.1.41 TCMBMEMTAB Register (Offset = 2C0h) [reset = 7654h]

TCMBMEMTAB is shown in Figure 4-807 and described in Table 4-844.

Return to Summary Table.

Controls ordering of banks in shared memory associated with MSS TCMB

Figure 4-807. TCMBMEMTAB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																TCMBMEMTAB															
0h																R/W-7654h															

Table 4-844. TCMBMEMTAB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU1		0h	Not Used
15-0	TCMBMEMTAB	R/W	7654h	MSS TCMB memory table for shared memory. Ordering of address in shared memory. 0th 128KB address goes to bank number programmed in [3:0] of this register(default is bank 4), 1st 128KB address goes to bank number programmed in [7:4] of this register(default is bank 5), 2nd 128KB address goes to bank number programmed in [11:8] of this register(default is bank 6), 3rd 128KB address goes to bank number programmed in [15:12] of this register(default is bank 7). Corresponding banks need to be selected for MSS TCMB in SHMEMBANKSEL regsiter

4.9.1.42 SHMEMBANKSEL3TO0 Register (Offset = 2C8h) [reset = 01010101h]

SHMEMBANKSEL3TO0 is shown in Figure 4-808 and described in Table 4-845.

Return to Summary Table.

Shared memory master allocation.Writing to each 8 bit field indicates the bank allocated to which master. 0x1 : DSS 0x2 : MSS TCMA 0x4 : MSS TCMB 0x8 : OCLA 0x10: BSS TCMA

Figure 4-808. SHMEMBANKSEL3TO0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BANK3								BANK2								BANK1								BANK0							
R/W-1h								R/W-1h								R/W-1h								R/W-1h							

Table 4-845. SHMEMBANKSEL3TO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BANK3	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
23-16	BANK2	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
15-8	BANK1	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)
7-0	BANK0	R/W	1h	only valid value is 0x1 else memory is not used. (Allocation only to DSS L3RAM)

4.9.1.43 SHMEMBANKSEL7TO4 Register (Offset = 2CCh) [reset = 01010101h]

SHMEMBANKSEL7TO4 is shown in Figure 4-809 and described in Table 4-846.

Return to Summary Table.

Shared memory master allocation.Writing to each 8 bit field indicates the bank allocated to which master. 0x1 : DSS 0x2 : MSS TCMA 0x4 : MSS TCMB 0x8 : OCLA 0x10: BSS TCMA

Figure 4-809. SHMEMBANKSEL7TO4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BANK7								BANK6								BANK5								BANK4							
R/W-1h								R/W-1h								R/W-1h								R/W-1h							

Table 4-846. SHMEMBANKSEL7TO4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BANK7	R/W	1h	only valid value is 0x1/0x2/0x4/0x8/0X10 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA,BSS TCMA)
23-16	BANK6	R/W	1h	only valid value is 0x1/0x2/0x4/0x8/0X10 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA,BSS TCMA)
15-8	BANK5	R/W	1h	only valid value is 0x1/0x2/0x4/0x8 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA)
7-0	BANK4	R/W	1h	only valid value is 0x1/0x2/0x4/0x8 else memory is not used. (Allocation to DSS L3RAM, MSS TCMA/B, OCLA)

4.9.1.44 PBISTCLKCTL Register (Offset = 2D0h) [reset = 1h]

PBISTCLKCTL is shown in Figure 4-810 and described in Table 4-847.

Return to Summary Table.

PBIST clock control register

Figure 4-810. PBISTCLKCTL Register

31	30	29	28	27	26	25	24
NU							
0h							
23	22	21	20	19	18	17	16
NU							
0h							
15	14	13	12	11	10	9	8
PBIST300MCLKGATE							
R/W-0h							
7	6	5	4	3	2	1	0
PBIST300MCLKSRCSEL				PBIST300MCLKDIV			
R/W-0h				R/W-1h			

Table 4-847. PBISTCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	Not used
15-8	PBIST300MCLKGATE	R/W	0h	Pre clock divider gate for PBIST300M clock. Gates the clock before divider. to gate the clock either 3:0 should be 0xD or 7:4 should be 0xA
7-4	PBIST300MCLKSRCSEL	R/W	0h	Select clock source for DSP PBIST Clock source 000 => CPUCLK 001, 100, 101, 111 => RCCLK 010 => 600-MHz PLL divided clock 110 => REFCLK from ANA
3-0	PBIST300MCLKDIV	R/W	1h	Divide value for DSP PBIST source clock selected by field PBIST300MCLKSRCSEL in register CLKSRCSEL. “0000” => div1 “0001” => div2 | | “1111” => div15 One Should change the divide value before switching to New clock. Switching to New clock is done by progrmming PBIST300MCLKSRCSEL

4.9.2 MSS_RCM Registers

Table 4-848 lists the memory-mapped registers for the MSS_RCM. All register offset addresses not listed in Table 4-848 should be considered as reserved locations and the register contents should not be modified.

Table 4-848. MSS_RCM Registers

Offset	Acronym	Register Name	Section
4h	SOFTRST1	SYS_SOFT_RESET1	Section 4.9.2.1
8h	SOFTRST2	SYS_SOFT_RESET2	Section 4.9.2.2
18h	CLKDIVCTL0	CLKDIV	Section 4.9.2.3
1Ch	CLKSRCSEL0	CLKSRCSEL	Section 4.9.2.4
20h	CR4CTL	CR4CTL	Section 4.9.2.5
3Ch	CLKGATE	CLK_GATE	Section 4.9.2.6
44h	CLKSRCSEL1	SYS_CLKSRCSEL	Section 4.9.2.7
54h	CURRCLKDIV0	CURR_CLKDIV	Section 4.9.2.8
58h	RTICURRCLKDIV	RTI_CURR_CLKDIV	Section 4.9.2.9
5Ch	MEMINITSTART	MEM_INIT_START	Section 4.9.2.10
60h	CURRCLKDIV1	CURR_CLKDIV1	Section 4.9.2.11
6Ch	MEMINITDONE	MEM_INIT_DONE	Section 4.9.2.12
70h	ECCENMSSGEM	ECCEN_MSSGEM	Section 4.9.2.13
74h	ECCCAPTMSSGEM	ECCCAPT_MSSGEM	Section 4.9.2.14
78h	ECCENBSSGEM	ECCEN_BSSGEM	Section 4.9.2.15
7Ch	ECCCAPTBSSGEM	ECCCAPT_BSSGEM	Section 4.9.2.16
80h	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.3.19
84h	NSYSPERUSERMODEN	NON_SYS_PERIPH_USERMODEEN	Section 4.9.2.18
88h	SECURERAMMMI	SECURERAMMMI	Section 4.9.2.19
8Ch	SECURERAMECC	SECURERAMECC	Section 4.9.2.20
90h	ESMGATE0	ESMGATE0	Section 4.9.2.21
94h	ESMGATE1	ESMGATE1	Section 4.9.2.22
98h	ESMGATE2	ESMGATE2	Section 4.9.2.23
9Ch	ESMGATE3	ESMGATE3	Section 4.9.2.24
A0h	ESMGATE4	ESMGATE4	Section 4.9.2.25
ACh	KEY	CFGREG_ACCESS_KEY	Section 4.9.2.26
B8h	SWIRQA	SWIRQ0	Section 4.9.2.27
BCh	SWIRQB	SWIRQ1	Section 4.9.2.28
C0h	MISCCTL0	MISCELLANEOUS_CTL_REG	Section 4.9.2.29
C4h	ATCMERRCAPTCTL	ATCMERRCAPT	Section 4.9.2.30
C8h	B0TCMERRCAPTCTL	B0TCMERRCAPT	Section 4.9.2.31
CCh	B1TCMERRCAPTCTL	B1TCMERRCAPT	Section 4.9.2.32
D0h	SOFTCORERST	SOFT_CORE_RST	Section 4.9.2.33
D8h	RSTCAUSE	MSS_RST_CAUSE	Section 4.9.2.34
DCh	RSTCAUSECLR	MSS_RST_CAUSE_CLR	Section 4.9.2.35
E0h	SPITRIGSRC	SPI_TRIG_SRC	Section 4.9.2.36
E4h	CLKINUSE	MSS_CLK_IN_USE	Section 4.9.2.37
E8h	ECCENMSSBSS	MSS_ECC_EN	Section 4.9.2.38
ECh	ECCCAPTMSSBSS	MSS_ECC_CAPT	Section 4.9.2.39
F0h	CLKDIVCTL2	CLKDIV2	Section 4.9.2.40
FCh	SWIRQC	SW_IRQ2	Section 4.9.2.41

Complex bit access types are encoded to fit into small table cells. Table 4-849 shows the codes that are used for access types in this section.

Table 4-849. MSS_RCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.9.2.1 SOFTRST1 Register (Offset = 4h) [reset = 0h]

SOFTRST1 is shown in Figure 4-811 and described in Table 4-850.

Return to Summary Table.

Figure 4-811. SOFTRST1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CR4SYSRST							
R/W-0h								0h							

Table 4-850. SOFTRST1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Reserved
7-0	CR4SYSRST		0h	Write 0xAD to assert a MSS CR4 only reset.Self clearing By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.

NOTE: This register used for the ROM eclipsing feature. Refer to the ROM Eclipsing section for more details.

4.9.2.2 SOFTRST2 Register (Offset = 8h) [reset = 0h]

SOFTRST2 is shown in Figure 4-812 and described in Table 4-851.

Return to Summary Table.

Figure 4-812. SOFTRST2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIMRST								RESERVED																							
R/W-0h								R/W-0h																							

Table 4-851. SOFTRST2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VIMRST	R/W	0h	Write 0xAD to assert a VIM only reset. By design reset will happen either lower 4 bit is 0XD or Upper four bit is 0xA.
23-0	RESERVED	R/W	0h	Reserved

4.9.2.3 CLKDIVCTL0 Register (Offset = 18h) [reset = 0h]

CLKDIVCTL0 is shown in Figure 4-813 and described in Table 4-852.

Return to Summary Table.

Figure 4-813. CLKDIVCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCANCLKDIV								RESERVED							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCLKDIV								RESERVED							
R/W-0h								R/W-0h							

Table 4-852. CLKDIVCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FDCANCLKDIV	R/W	0h	Divide value for FDCAN source clock selected by field FDCANCLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
23-16	RESERVED	R/W	0h	Reserved
15-8	VCLKCLKDIV	R/W	0h	Divide value for MSS subsystem source clock selected by field VCLKCLKSRCSEL in register CLKSRCSEL1 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256
7-0	RESERVED	R/W	0h	Reserved

4.9.2.4 CLKSRCSEL0 Register (Offset = 1Ch) [reset = 0h]

CLKSRCSEL0 is shown in Figure 4-814 and described in Table 4-853.

Return to Summary Table.

Figure 4-814. CLKSRCSEL0 Register

31	30	29	28	27	26	25	24
RTIDCLKSRCSEL				RTICCLKSRCSEL			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
RESERVED				QSPICLKSRCSEL			
0h				R/W-0h			
15	14	13	12	11	10	9	8
RESERVED				FDCANCLKSRCSEL			
0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED							
0h							

Table 4-853. CLKSRCSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RTIDCLKSRCSEL	R/W	0h	Select clock source for RTID baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
27-24	RTICCLKSRCSEL	R/W	0h	Select clock source for RTIC baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
23-20	RESERVED		0h	
19-16	QSPICLKSRCSEL	R/W	0h	Select clock source for QSPI baud clock 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA (40Mhz or 50 Mhz or 80Mh or 100Mh) 111 => RCCLK
15-12	RESERVED		0h	
11-8	FDCANCLKSRCSEL	R/W	0h	Select clock source for FDCANCLKSRCSEL 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK from ANA 111 => RCCLK
7-0	RESERVED		0h	

4.9.2.5 CR4CTL Register (Offset = 20h) [reset = 0h]

CR4CTL is shown in Figure 4-815 and described in Table 4-854.

Return to Summary Table.

Figure 4-815. CR4CTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU								MEMSWAPWAIT							
0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR4MEMSWAP								RESERVED							
R/W-0h								R/W-0h							

Table 4-854. CR4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU		0h	Reserved
23-16	MEMSWAPWAIT	R/W	0h	When CR4MEMSWAP is 0xAD : Write 0xAD to this field enable the CR4MEMSWAP only after a CR4 reset – either by writing to CR4SYSRST or by writing to PRCR register in CR4 debug space.
15-8	CR4MEMSWAP	R/W	0h	Write 0xAD will map the MSS CR4 0x0000_0000 to MSS CR4 TCMA RAM start address.
7-0	RESERVED	R/W	0h	Reserved

4.9.2.6 CLKGATE Register (Offset = 3Ch) [reset = 400h]

CLKGATE is shown in Figure 4-816 and described in Table 4-855.

Return to Summary Table.

Figure 4-816. CLKGATE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RTIDCLKGATE	RTICCLKGATE	FDCANCLKGATE	RESERVED	QSPICLKGATE	RESERVED		
R/W-0h	R/W-0h	R/W-0h	0h	R/W-0h	0h		

Table 4-855. CLKGATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED		0h	Reserved
7	RTIDCLKGATE	R/W	0h	Pre clock divider gate for RTID clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
6	RTICCLKGATE	R/W	0h	Pre clock divider gate for RTIC clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
5	FDCANCLKGATE	R/W	0h	Pre clock divider gate for FDCAN clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
4	RESERVED	R/W	0h	Reserved
3	QSPICLKGATE	R/W	0h	Pre clock divider gate for QSPI clock. Gates the clock before divider. Gates the clock before divider. 1: Gate the clock. 0: Ungate the clock.
2-0	RESERVED	R/W	0h	Reserved

4.9.2.7 CLKSRCSEL1 Register (Offset = 44h) [reset = 0h]

CLKSRCSEL1 is shown in Figure 4-817 and described in Table 4-856.

Return to Summary Table.

Figure 4-817. CLKSRCSEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED							
0h							
7	6	5	4	3	2	1	0
RESERVED				VCLKCLKSRCSEL			
0h				R/W-0h			

Table 4-856. CLKSRCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED		0h	Reserved
3-0	VCLKCLKSRCSEL	R/W	0h	Select clock source for MSS subsystem VCLK 000 => CPUCLK 00, 100, 101, 111 => RCCLK 010 => 600-MHz PLL divided clock 100 => CPUCLK 110 => REFCLK from ANA

4.9.2.8 CURRCLKDIV0 Register (Offset = 54h) [reset = 0h]

CURRCLKDIV0 is shown in Figure 4-818 and described in Table 4-857.

Return to Summary Table.

Figure 4-818. CURRCLKDIV0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCANCURRCLKDIV								RESERVED							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCLKCURRCLKDIV								RESERVED							
R-0h								0h							

Table 4-857. CURRCLKDIV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	FDCANCURRCLKDIV	R	0h	Returns Current divide value of FDCAN baud clock divider.
23-16	RESERVED	R	0h	Reserved
15-8	VCLKCURRCLKDIV	R	0h	Returns Current divide value of VCLK divider.
7-0	RESERVED		0h	Reserved

4.9.2.9 RTICURRCLKDIV Register (Offset = 58h) [reset = 0h]

RTICURRCLKDIV is shown in Figure 4-819 and described in Table 4-858.

Return to Summary Table.

Figure 4-819. RTICURRCLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU								NU							
0h								0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTCURRCLKDIV								RTICURRCLKDIV							
R-0h								R-0h							

Table 4-858. RTICURRCLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU		0h	
15-8	WDTCURRCLKDIV	R	0h	Not used.Reserved
7-0	RTICURRCLKDIV	R	0h	Not used.Reserved

4.9.2.10 MEMINITSTART Register (Offset = 5Ch) [reset = 0h]

MEMINITSTART is shown in Figure 4-820 and described in Table 4-859.

Return to Summary Table.

Figure 4-820. MEMINITSTART Register

31	30	29	28	27	26	25	24
MEMINITKEY							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED						BSSMBOX4GEMMEM	MSSMBOX4GEMMEM
0h						0h	0h
15	14	13	12	11	10	9	8
GEMMBOX4MSSMEM	GEMMBOX4BSSMEM	RESERVED				DMA2MEM	BSSMBOX4MSSMEM
0h	0h	0h				0h	0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	RESERVED	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-859. MEMINITSTART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MEMINITKEY	R/W	0h	Memory hardware initialization global enable key. Write 0XAD to enable MEMINIT.
23-18	RESERVED		0h	Reserved
17	BSSMBOX4GEMMEM		0h	Writing '1' will trigger DSS- BSS mailbox initialization.Self clearing
16	MSSMBOX4GEMMEM		0h	Writing '1' will trigger DSS- MSS mailbox initialization.Self clearing
15	GEMMBOX4MSSMEM		0h	Writing '1' will trigger DSS- MSS mailbox initialization.Self clearing
14	GEMMBOX4BSSMEM		0h	Writing '1' will trigger DSS- BSS mailbox initialization.Self clearing
13-10	RESERVED		0h	Reserved
9	DMA2MEM		0h	Writing '1' will trigger DMA2 memory initialization.Self clearing
8	BSSMBOX4MSSMEM		0h	Writing '1' will trigger MSS- BSS mailbox initialization.Self clearing
7	MSSMBOX4BSSMEM		0h	Writing '1' will trigger MSS- BSS mailbox initialization.Self clearing
6	RESERVED		0h	Reserved
5	SPIBMEM		0h	Writing '1' will trigger SPIB memory initialization.Self clearing
4	SPIAMEM		0h	Writing '1' will trigger SPIA memory initialization.Self clearing
3	VIMMEM		0h	Writing '1' will trigger VIM memory initialization.Self clearing
2	DMAMEM		0h	Writing '1' will trigger DMA memory initialization.Self clearing
1	CR4TCMBMEM		0h	Writing '1' will trigger MSS TCMB memory initialization.Self clearing
0	CR4TCMAMEM		0h	Writing '1' will trigger MSS TCMA memory initialization.Self clearing

4.9.2.11 CURRCLKDIV1 Register (Offset = 60h) [reset = 0h]

CURRCLKDIV1 is shown in Figure 4-821 and described in Table 4-860.

Return to Summary Table.

Figure 4-821. CURRCLKDIV1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								QSPICURRCLKDIV							
R-0h								R-0h							

Table 4-860. CURRCLKDIV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	QSPICURRCLKDIV	R	0h	Returns Current divide value of QSPI_CLK divider.

4.9.2.12 MEMINITDONE Register (Offset = 6Ch) [reset = 0h]

MEMINITDONE is shown in Figure 4-822 and described in Table 4-861.

Return to Summary Table.

Figure 4-822. MEMINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED						BSSMBOX4GEMMEM	MSSMBOX4GEMMEM
0h						R-0h	R-0h
15	14	13	12	11	10	9	8
GEMMBOX4MSSMEM	GEMMBOX4BSSMEM	RESERVED				DMA2MEM	BSSMBOX4MSSMEM
R-0h	R-0h	R-0h				R-0h	R-0h
7	6	5	4	3	2	1	0
MSSMBOX4BSSMEM	RESERVED	SPIBMEM	SPIAMEM	VIMMEM	DMAMEM	CR4TCMBMEM	CR4TCMAMEM
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 4-861. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED		0h	Reserved
17	BSSMBOX4GEMMEM	R	0h	Memory Initialization done status for DSS- BSS mailbox
16	MSSMBOX4GEMMEM	R	0h	Memory Initialization done status for DSS- MSS mailbox
15	GEMMBOX4MSSMEM	R	0h	Memory Initialization done status for DSS- MSS mailbox
14	GEMMBOX4BSSMEM	R	0h	Memory Initialization done status for DSS- BSS mailbox
13-10	RESERVED	R	0h	Reserved
9	DMA2MEM	R	0h	Memory Initialization done status for MSS DMA2 memory
8	BSSMBOX4MSSMEM	R	0h	Memory Initialization done status for MSS- BSS mailbox
7	MSSMBOX4BSSMEM	R	0h	Memory Initialization done status for MSS- BSS mailbox
6	RESERVED	R	0h	Reserved
5	SPIBMEM	R	0h	Memory Initialization done status for MSS SPIB memory
4	SPIAMEM	R	0h	Memory Initialization done status for MSS SPIA memory
3	VIMMEM	R	0h	Memory Initialization done status for MSS VIM memory
2	DMAMEM	R	0h	Memory Initialization done status for MSS DMA memory
1	CR4TCMBMEM	R	0h	Memory Initialization done status for MSS TCMB memory
0	CR4TCMAMEM	R	0h	Memory Initialization done status for MSS TCMA memory

4.9.2.13 ECCENMSSGEM Register (Offset = 70h) [reset = 0h]

ECCENMSSGEM is shown in Figure 4-823 and described in Table 4-862.

Return to Summary Table.

Figure 4-823. ECCENMSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENMSSGEM																															
R/W-0h																															

Table 4-862. ECCENMSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENMSSGEM	R/W	0h	7:0 : Writing 0xAD will enable ECC for MSS- DSS mailbox 15:8: Writing 0xAD will enable ECC for MSS- DSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in MSS mailbox for GEM 21:19: Write 3'b111 to clear the Address captured because of ECC error. In GEM mailbox for MSS

4.9.2.14 ECCCAPTMSSGEM Register (Offset = 74h) [reset = 0h]

ECCCAPTMSSGEM is shown in Figure 4-824 and described in Table 4-863.

Return to Summary Table.

Figure 4-824. ECCCAPTMSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTMSSGEM																															
R-0h																															

Table 4-863. ECCCAPTMSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTMSSGEM	R	0h	7:0 : mss_mbox4gem_ecc_fault_address 14:8 : mss_mbox4gem_repaired_bit 23:16 : gem_mbox4mss_ecc_fault_address 30:24 : gem_mbox4mss_repaired_bit

4.9.2.15 ECCENBSSGEM Register (Offset = 78h) [reset = 0h]

ECCENBSSGEM is shown in Figure 4-825 and described in Table 4-864.

Return to Summary Table.

Figure 4-825. ECCENBSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENBSSGEM																															
R/W-0h																															

Table 4-864. ECCENBSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENBSSGEM	R/W	0h	7:0 : Writing 0xAD will enable ECC for DSS- BSS mailbox 15:8: Writing 0xAD will enable ECC for DSS- BSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in GEM mailbox for BSS 21:19: Write 3'b111 to clear the Address captured because of ECC error in BSS mailbox for GEM

4.9.2.16 ECCCAPTBSSGEM Register (Offset = 7Ch) [reset = 0h]

ECCCAPTBSSGEM is shown in Figure 4-826 and described in Table 4-865.

Return to Summary Table.

Figure 4-826. ECCCAPTBSSGEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTBSSGEM																															
R-0h																															

Table 4-865. ECCCAPTBSSGEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTBSSGEM	R	0h	7:0 : bss_mbox4gem_ecc_fault_address 14:8 : bss_mbox4gem_repaired_bit 23:16 : gem_mbox4bss_ecc_fault_address 30:24 : gem_mbox4bss_repaired_bit

4.9.2.17 USERMODEEN Register (Offset = 80h) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-827. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-866. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS RCM space.

4.9.2.18 NSYSPERUSERMODEN Register (Offset = 84h) [reset = 0h]

NSYSPERUSERMODEN is shown in Figure 4-828 and described in Table 4-867.

Return to Summary Table.

Figure 4-828. NSYSPERUSERMODEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSYSPERUSERMODEN																															
R/W-0h																															

Table 4-867. NSYSPERUSERMODEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NSYSPERUSERMODEN	R/W	0h	2:0 : Write 3'b111 to enable user mode access to SPIA 5:3 : Write 3'b111 to enable user mode access to SPIB 10:8 : Write 3'b111 to enable user mode access to GIO 13:11 : Write 3'b111 to enable user mode access to QSPI 18:16 : Write 3'b111 to enable user mode access to SCIA 21:19 : Write 3'b111 to enable user mode access to SCIB 26:24

4.9.2.19 SECURERAMMMI Register (Offset = 88h) [reset = 0h]

SECURERAMMMI is shown in Figure 4-829 and described in Table 4-868.

Return to Summary Table.

Figure 4-829. SECURERAMMMI Register

31	30	29	28	27	26	25	24
NU2							
0h							
23	22	21	20	19	18	17	16
NU2							SECURERAMINITDONE
0h							R-0h
15	14	13	12	11	10	9	8
NU1							
0h							
7	6	5	4	3	2	1	0
NU1							SECURERAMINIT
0h							0h

Table 4-868. SECURERAMMMI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU2		0h	
16	SECURERAMINITDONE	R	0h	Memory Initialization done status for Secure Key RAM
15-1	NU1		0h	
0	SECURERAMINIT		0h	Writing '1' will trigger Secure Key RAM memory initialization.Self clearing

4.9.2.20 SECURERAMECC Register (Offset = 8Ch) [reset = 0h]

SECURERAMECC is shown in Figure 4-830 and described in Table 4-869.

Return to Summary Table.

Figure 4-830. SECURERAMECC Register

31	30	29	28	27	26	25	24
SECURERAMBIT							
R-0h							
23	22	21	20	19	18	17	16
SECURERAMADDR							
R-0h							
15	14	13	12	11	10	9	8
NU					SECURERAMECCCLR		
0h					R/W-0h		
7	6	5	4	3	2	1	0
SECURERAMECCEN							
R/W-0h							

Table 4-869. SECURERAMECC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SECURERAMBIT	R	0h	Secure key RAM repaired bit
23-16	SECURERAMADDR	R	0h	Secure Key RAM_ecc_fault_address
15-11	NU		0h	
10-8	SECURERAMECCCLR	R/W	0h	10:8 : Write 3'b111 to clear the Address captured because of ECC error
7-0	SECURERAMECCEN	R/W	0h	7:0 : Writing 0xAD will enable ECC for Secure key RAM

4.9.2.21 ESMGATE0 Register (Offset = 90h) [reset = 0h]

ESMGATE0 is shown in Figure 4-831 and described in Table 4-870.

Return to Summary Table.

Figure 4-831. ESMGATE0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE0																															
R/W-07070700h																															

Table 4-870. ESMGATE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE0	R/W	07070700h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 0 [7:4] : Gate ESM group2 line 1 [11:8] : Gate ESM group2 line 2 [15:12] : Gate ESM group2 line 3 [19:16] : Gate ESM group2 line 4 [23:20] : Gate ESM group2 line 5 [27:24] : Gate ESM group2 line 6 [31:28] : Gate ESM group2 line 7 Static register setting.Should not be changed on the fly

4.9.2.22 ESMGATE1 Register (Offset = 94h) [reset = 0h]

ESMGATE1 is shown in Figure 4-832 and described in Table 4-871.

Return to Summary Table.

Figure 4-832. ESMGATE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE1																															
R/W-7h																															

Table 4-871. ESMGATE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE1	R/W	7h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 8 [7:4] : Gate ESM group2 line 9 [11:8] : Gate ESM group2 line 10 [15:12] : Gate ESM group2 line 11 [19:16] : Gate ESM group2 line 12 [23:20] : Gate ESM group2 line 13 [27:24] : Gate ESM group2 line 14 [31:28] : Gate ESM group2 line 15 Static register setting.Should not be changed on the fly

4.9.2.23 ESMGATE2 Register (Offset = 98h) [reset = 0h]

ESMGATE2 is shown in Figure 4-833 and described in Table 4-872.

Return to Summary Table.

Figure 4-833. ESMGATE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE2																															
R/W-7h																															

Table 4-872. ESMGATE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE2	R/W	7h	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23 Static register setting should not be changed on the fly

4.9.2.24 ESMGATE3 Register (Offset = 9Ch) [reset = 0h]

ESMGATE3 is shown in Figure 4-834 and described in Table 4-873.

Return to Summary Table.

Figure 4-834. ESMGATE3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE3																															
R/W-3Fh																															

Table 4-873. ESMGATE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE3	R/W	3Fh	write 4'b111 to Gate [3:0] : Gate ESM group2 line 16 [7:4] : Gate ESM group2 line 17 [11:8] : Gate ESM group2 line 18 [15:12] : Gate ESM group2 line 19 [19:16] : Gate ESM group2 line 20 [23:20] : Gate ESM group2 line 21 [27:24] : Gate ESM group2 line 22 [31:28] : Gate ESM group2 line 23 Static register setting.Should not be changed on the fly

4.9.2.25 ESMGATE4 Register (Offset = A0h) [reset = 0h]

ESMGATE4 is shown in Figure 4-835 and described in Table 4-874.

Return to Summary Table.

Figure 4-835. ESMGATE4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGATE4																															
R/W-0h																															

Table 4-874. ESMGATE4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGATE4	R/W	0h	write 4'b111 to Gate [3:0] : Gate ESM group3 line 0 [7:4] : Gate ESM group3 line 1 [11:8] : Gate ESM group3 line 2 [15:12] : Gate ESM group3 line 3 [19:16] : Gate ESM group3 line 4 [23:20] : Gate ESM group3 line 5 [27:24] : Gate ESM group3 line 6 [31:28] : Gate ESM group3 line 7 Static register setting.Should not be changed on the fly

4.9.2.26 KEY Register (Offset = ACh) [reset = 83E783E7h]

KEY is shown in Figure 4-836 and described in Table 4-875.

Return to Summary Table.

Figure 4-836. KEY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY																															
R/W-83E783E7h																															

Table 4-875. KEY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY	R/W	83E783E7h	Kicker Register. The value 83E7_83E7h must be written as part of the process to unlock the CPU write access to the MSS RCM registers

4.9.2.27 SWIRQA Register (Offset = B8h) [reset = 0h]

SWIRQA is shown in Figure 4-837 and described in Table 4-876.

Return to Summary Table.

Figure 4-837. SWIRQA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ1								SWIRQ1DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ0								SWIRQ0DAT							
R/W-0h								R/W-0h							

Table 4-876. SWIRQA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ1	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ1DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ0	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ0DAT	R/W	0h	Not Used.Reserved

4.9.2.28 SWIRQB Register (Offset = BCh) [reset = 0h]

SWIRQB is shown in Figure 4-838 and described in Table 4-877.

Return to Summary Table.

Figure 4-838. SWIRQB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ3								SWIRQ3DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ2								SWIRQ2DAT							
R/W-0h								R/W-0h							

Table 4-877. SWIRQB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ3	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ3DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ2	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ2DAT	R/W	0h	Not Used.Reserved

4.9.2.29 MISCCTL0 Register (Offset = C0h) [reset = ADADh]

MISCCTL0 is shown in Figure 4-839 and described in Table 4-878.

Return to Summary Table.

Figure 4-839. MISCCTL0 Register

31	30	29	28	27	26	25	24
NU							TCMB1EZDIS
0h							R/W-0h
23	22	21	20	19	18	17	16
TCMB1EZDIS			TCMB0EZDIS			TCMAEZDIS	
R/W-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							
R/W-ADh							
7	6	5	4	3	2	1	0
RESERVED							
R/W-ADh							

Table 4-878. MISCCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU		0h	
24-22	TCMB1EZDIS	R/W	0h	Write 111 to make TCMB1 EZ to '1' regardless of Functional value.
21-19	TCMB0EZDIS	R/W	0h	Write 111 to make TCMB0 EZ to '1' regardless of Functional value.
18-16	TCMAEZDIS	R/W	0h	Write 111 to make TCMA EZ to '1' regardless of Functional value.
15-0	RESERVED	R/W	ADh	Reserved

4.9.2.30 ATCMERRCAPTCTL Register (Offset = C4h) [reset = 0h]

ATCMERRCAPTCTL is shown in Figure 4-840 and described in Table 4-879.

Return to Summary Table.

Figure 4-840. ATCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRATCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRATCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRATCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		ATCFORCEERR			ERRATCADDRCLR		
0h		R/W-0h			0h		

Table 4-879. ATCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRATCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	ATCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRATCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.9.2.31 B0TCMERRCAPTCTL Register (Offset = C8h) [reset = 0h]

B0TCMERRCAPTCTL is shown in Figure 4-841 and described in Table 4-880.

Return to Summary Table.

Figure 4-841. B0TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB0TCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRB0TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB0TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B0TCFORCEERR			ERRB0TCADDRCLR		
0h		R/W-0h			0h		

Table 4-880. B0TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRB0TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	B0TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB0TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.9.2.32 B1TCMERRCAPTCTL Register (Offset = CCh) [reset = 0h]

B1TCMERRCAPTCTL is shown in Figure 4-842 and described in Table 4-881.

Return to Summary Table.

Figure 4-842. B1TCMERRCAPTCTL Register

31	30	29	28	27	26	25	24
NU1				ERRB1TCADDR			
0h				R-0h			
23	22	21	20	19	18	17	16
ERRB1TCADDR							
R-0h							
15	14	13	12	11	10	9	8
ERRB1TCADDR							
R-0h							
7	6	5	4	3	2	1	0
NU0		B1TCFORCEERR			ERRB1TCADDRCLR		
0h		R/W-0h			0h		

Table 4-881. B1TCMERRCAPTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU1		0h	Reserved
27-8	ERRB1TCADDR	R	0h	TCM address for which parity error happened
7-6	NU0		0h	Reserved
5-3	B1TCFORCEERR	R/W	0h	Write 111 to force error in TCM address control parity check logic.
2-0	ERRB1TCADDRCLR		0h	When parity error happens on TCM address control path, Latch that captures the address for which error happened is disabled. Write 111 to re-enable the latching.Self clearing

4.9.2.33 SOFTCORERST Register (Offset = D0h) [reset = 000F0F00h]

SOFTCORERST is shown in Figure 4-843 and described in Table 4-882.

Return to Summary Table.

Figure 4-843. SOFTCORERST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RST_WFICHECKEN								RESERVED							
R/W-0h								R/W-Fh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSTTOASSRTDLY								RESERVED							
R/W-Fh								0h							

Table 4-882. SOFTCORERST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RST_WFICHECKEN	R/W	0h	0xAD : When CR4SYSRST is set, Before asserting the reset to CR4 wait for CR4 to enter WFI. 0X0: Dont wait for WFI
23-16	RESERVED	R/W	Fh	Reserved
15-8	RSTTOASSRTDLY	R/W	Fh	Wait for programmed number of clock cycle before reset is asserted to CR4.
7-0	RESERVED		0h	Reserved

4.9.2.34 RSTCAUSE Register (Offset = D8h) [reset = 0h]

RSTCAUSE is shown in Figure 4-844 and described in Table 4-883.

Return to Summary Table.

Figure 4-844. RSTCAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								RSTCAUSE							
0h																								R-0h							

Table 4-883. RSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	Reserved
7-0	RSTCAUSE	R	0h	MSS_RCM:RSTCAUSE 0000_1001 : System out of NRESET 0000_1000 : System out of Warm Reset 0010_0000 : CR4 reset because of Software trigger. 0001_0000 : CR4 STC reset 0100_0000 : CR4 reset because of writing to PRCR register in CR4 debug space. Note: Because the ROM Bootloader issues a soft reset to pass control to the application (or secondary bootloader), this register always reads 0x20. The reset value is stored by the ROM Bootloader into TOPRCM_SPARE9. Refer to that register description to get the actual power-on or reset value.

4.9.2.35 RSTCAUSECLR Register (Offset = DCh) [reset = 0h]

RSTCAUSECLR is shown in Figure 4-845 and described in Table 4-884.

Return to Summary Table.

Figure 4-845. RSTCAUSECLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								RSTCAUSECLR							
0h								0h							

Table 4-884. RSTCAUSECLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	
7-0	RSTCAUSECLR		0h	Write 0xAD to clear RSTCAUSE.Self clearing

4.9.2.36 SPITRIGSRC Register (Offset = E0h) [reset = 0h]

SPITRIGSRC is shown in Figure 4-846 and described in Table 4-885.

Return to Summary Table.

Figure 4-846. SPITRIGSRC Register

31	30	29	28	27	26	25	24
NU3					SPIBTRIG		
0h					R/W-0h		
23	22	21	20	19	18	17	16
SPIBTRIG							
R/W-0h							
15	14	13	12	11	10	9	8
NU2							SPIATRIG1
0h							R/W-0h
7	6	5	4	3	2	1	0
NU1							SPIATRIG0
0h							R/W-0h

Table 4-885. SPITRIGSRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU3		0h	
26-16	SPIBTRIG	R/W	0h	[20:16] --> Trigger sources for MIBSPIB
15-9	NU2		0h	
8	SPIATRIG1	R/W	0h	1st bit ofTRG_SRC input of SPIA.
7-1	NU1		0h	
0	SPIATRIG0	R/W	0h	0th bit ofTRG_SRC input of SPIA.

4.9.2.37 CLKINUSE Register (Offset = E4h) [reset = 0h]

CLKINUSE is shown in Figure 4-847 and described in Table 4-886.

Return to Summary Table.

Figure 4-847. CLKINUSE Register

31	30	29	28	27	26	25	24
RESERVED				FRCCLKINUSE			
0h				R-0h			
23	22	21	20	19	18	17	16
RTIDCLKINUSE				RTICCLKINUSE			
R-0h				R-0h			
15	14	13	12	11	10	9	8
QSPICLKINUSE				RESERVED			
R-0h				0h			
7	6	5	4	3	2	1	0
FDCANCLKINUSE				VCLKINUSE			
R-0h				R-0h			

Table 4-886. CLKINUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED		0h	Reserved
27-24	FRCCLKINUSE	R	0h	Current Clock Source Select Mux value for FRC CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
23-20	RTIDCLKINUSE	R	0h	Current Clock Source Select Mux value for RTID 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
19-16	RTICCLKINUSE	R	0h	Current Clock Source Select Mux value for RTIC 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
15-12	QSPICLKINUSE	R	0h	Current Clock Source Select Mux value for QSPI CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
11-8	RESERVED	R	0h	Reserved
7-4	FDCANCLKINUSE	R	0h	Current Clock Source Select Mux value for FDCAN CLK 000 => VCLK 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 100 => CPUCLK 101 => RCCLK 110 => REFCLK 111 => RCCLK (10Mhz)
3-0	VCLKINUSE	R	0h	Current Clock Source Select Mux value for VCLK 000 => CPUCLK (40Mhz or 50 Mhz or 80Mh or 100Mh) 001 => RCCLK (10Mhz) 010 => 600Mhz PLL divided clock 011 => 240Mhz PLL divided clock 101 => RCCLK (10Mhz) 110 => REFCLK 111 => RCCLK (10Mhz)

4.9.2.38 ECCENMSSBSS Register (Offset = E8h) [reset = 0h]

ECCENMSSBSS is shown in Figure 4-848 and described in Table 4-887.

Return to Summary Table.

Figure 4-848. ECCENMSSBSS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCENMSSBSS																															
R/W-0h																															

Table 4-887. ECCENMSSBSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCENMSSBSS	R/W	0h	7:0 : Writing 0xAD will enable ECC for MSS – BSS mailbox 15:8: Writing 0xAD will enable ECC for MSS – BSS mailbox 18:16 : Write 3'b111 to clear the Address captured because of ECC error in MSS mailbox for BSS 21:19: Write 3'b111 to clear the Address captured because of ECC error. In BSS mailbox for MSS

4.9.2.39 ECCCAPTMSSBSS Register (Offset = ECh) [reset = 0h]

ECCCAPTMSSBSS is shown in Figure 4-849 and described in Table 4-888.

Return to Summary Table.

Figure 4-849. ECCCAPTMSSBSS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCAPTMSSBSS																															
R-0h																															

Table 4-888. ECCCAPTMSSBSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECCCAPTMSSBSS	R	0h	7:0 : mss_mbox4bss_ecc_fault_address 14:8 : mss_mbox4bss_repaired_bit 23:16 : bss_mbox4mss_ecc_fault_address 30:24 : bss_mbox4mss_repaired_bit 31 : Reserved

4.9.2.40 CLKDIVCTL2 Register (Offset = F0h) [reset = 0h]

CLKDIVCTL2 is shown in Figure 4-850 and described in Table 4-889.

Return to Summary Table.

Figure 4-850. CLKDIVCTL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								QSPICLKDIV							
0h								R/W-0h							

Table 4-889. CLKDIVCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU		0h	
7-0	QSPICLKDIV	R/W	0h	Divide value for QSPI baud clock selected by field QSPICLKSRCSEL in register CLKSRCSEL0 0000_0000 => div1 0000_0001 => div2 | | 1111_1111 => div256

4.9.2.41 SWIRQC Register (Offset = FCh) [reset = 0h]

SWIRQC is shown in Figure 4-851 and described in Table 4-890.

Return to Summary Table.

Figure 4-851. SWIRQC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIRQ5								SWIRQ5DAT							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIRQ4								SWIRQ4DAT							
R/W-0h								R/W-0h							

Table 4-890. SWIRQC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SWIRQ5	R/W	0h	Write 0XAD to trigger interrupt.
23-16	SWIRQ5DAT	R/W	0h	Not Used.Reserved
15-8	SWIRQ4	R/W	0h	Write 0XAD to trigger interrupt.
7-0	SWIRQ4DAT	R/W	0h	Not Used.Reserved

4.9.3 MSS_GPCFG_REG Registers

Table 4-891 lists the memory-mapped registers for the MSS_GPCFG_REG. All register offset addresses not listed in Table 4-891 should be considered as reserved locations and the register contents should not be modified.

Table 4-891. MSS_GPCFG_REG Registers

Offset	Acronym	Register Name	Section
0h	GPCFG0	GPCFG0	Section 4.9.3.1
4h	GPCFG1	GPCFG1	Section 4.9.3.2
8h	GPCFG2	GPCFG2	Section 4.9.3.3
Ch	GPCFG3	GPCFG3	Section 4.9.3.4
10h	GPCFG4	GPCFG4	Section 4.9.3.5
2Ch	GPCFG11	GPCFG11	Section 4.9.3.7
D0h	CCCACFG0	CCCA_CFG0	Section 4.9.3.8
D4h	CCCACFG1	CCCA_CFG1	Section 4.9.3.9
D8h	CCCACFG2	CCCA_CFG2	Section 4.9.3.10
DCh	CCCACFG3	CCCA_CFG3	Section 4.9.3.11
E0h	CCCBCFG0	CCCB_CFG0	Section 4.9.3.12
E4h	CCCBCFG1	CCCB_CFG1	Section 4.9.3.13
E8h	CCCBCFG2	CCCB_CFG2	Section 4.9.3.14
ECh	CCCBCFG3	CCCB_CFG3	Section 4.9.3.15
F0h	CCCACNTVAL	CCCACNTVAL	Section 4.9.3.16
F4h	CCCBCNTVAL	CCCBCNTVAL	Section 4.9.3.17
F8h	CCCABERRSTAT	CCCABERRSTAT	Section 4.9.3.18
FCh	USERMODEEN	USER_MODE_ACCESS_EN	Section 4.9.3.19
140h	EPWMCFG	EPWMCFG	Section 4.9.3.20
148h	DMMSWINT0	DMMSWINT0	Section 4.9.3.21
14Ch	DMMSWINT1	DMMSWINT1	Section 4.9.3.22
150h	DMMSWINTSEL0	DMMSWINTSEL0	Section 4.9.3.23
154h	DMMSWINTSEL1	DMMSWINTSEL1	Section 4.9.3.24
158h	CCCBWDEN	CCCBWDEN	Section 4.9.3.25
15Ch	GPIOINTREDGESEL	GPIOINTREDGESEL	Section 4.9.3.26
164h	PWMDMATRIGEN	PWMDMATRIGEN	Section 4.9.3.27
168h	JTAGTXDATA	JTAGTXDATA	Section 4.9.3.28
16Ch	JTAGTXCONTROL	JTAGTXCONTROL	Section 4.9.3.29
170h	JTAGRXDATA	JTAGRXDATA	Section 4.9.3.30
178h	JTAGTXRXACK	JTAGTXRXACK	Section 4.9.3.31
17Ch	JTAGRXCONTROL	JTAGRXCONTROL	Section 4.9.3.32
180h	MSS2GEMSWIRQ		Section 4.9.3.33
184h	CSETBFLUSH		Section 4.9.3.34

Complex bit access types are encoded to fit into small table cells. Table 4-892 shows the codes that are used for access types in this section.

Table 4-892. MSS_GPCFG_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write

Table 4-892. MSS_GPCFG_REG Access Type Codes (continued)

Access Type	Code	Description
Reset or Default Value		
-n		Value after reset or the default value

4.9.3.1 GPCFG0 Register (Offset = 0h) [reset = 0h]

GPCFG0 is shown in Figure 4-852 and described in Table 4-893.

Return to Summary Table.

Figure 4-852. GPCFG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG0																															
R/W-0h																															

Table 4-893. GPCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG0	R/W	0h	General Purpose config register for SW use.

4.9.3.2 GPCFG1 Register (Offset = 4h) [reset = 0h]

GPCFG1 is shown in Figure 4-853 and described in Table 4-894.

Return to Summary Table.

Figure 4-853. GPCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG1																															
R/W-0h																															

Table 4-894. GPCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG1	R/W	0h	General Purpose config register for SW use.

4.9.3.3 GPCFG2 Register (Offset = 8h) [reset = 0h]

GPCFG2 is shown in Figure 4-854 and described in Table 4-895.

Return to Summary Table.

Figure 4-854. GPCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG2																															
R/W-0h																															

Table 4-895. GPCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG2	R/W	0h	General Purpose config register for SW use.

4.9.3.4 GPCFG3 Register (Offset = Ch) [reset = 0h]

GPCFG3 is shown in Figure 4-855 and described in Table 4-896.

Return to Summary Table.

Figure 4-855. GPCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG3																															
R/W-0h																															

Table 4-896. GPCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG3	R/W	0h	General Purpose config register for SW use.

4.9.3.5 GPCFG4 Register (Offset = 10h) [reset = 0h]

GPCFG4 is shown in Figure 4-856 and described in Table 4-897.

Return to Summary Table.

Figure 4-856. GPCFG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG4																															
R/W-0h																															

Table 4-897. GPCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG4	R/W	0h	General Purpose config register for SW use.

4.9.3.6 GPCFG6 Register (Offset = 18h) [reset = 0h]

GPCFG6 is shown in Figure 4-857 and described in Table 4-898.

Return to Summary Table.

Figure 4-857. GPCFG6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPCFG6																															
R/W-0h																															

Table 4-898. GPCFG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPCFG6	R/W	0h	Describes the muxing of Interrupts and DMA based on use cases - GPCFG6[0] = 1 : dma_req[36] = can_fd_intr[0] ,0 : GPCFG6[0] = 0 : dma_req[36] = can_fd_fe_intr[0] GPCFG6[1] = 1 : dma_req[44] = can_fd_intr[1] ,0 : GPCFG6[1] = 0 : dma_req[44] = can_fd_fe_intr[1] GPCFG6[2] = 1 : frc_can_intr[3] = can_fd_fe_intr[0] ,0 : GPCFG6[2] = 0 : frc_can_intr[3] = can_fd_intr[0] GPCFG6[3] = 1 : frc_can_intr[2] = can_fd_fe_intr[1] ,0 : GPCFG6[3] = 0 : frc_can_intr[2] = can_fd_intr[1] GPCFG6[8] = 1 : dma_req[5] = can_fd_fe_intr[2] ,0 : GPCFG6[8] = 0 : dma_req[5] = spia_dma_req[3] GPCFG6[11] = 1 : dma_req[37] = rti2 dma req[0] ,0 : GPCFG6[11] = 0 : dma_req[37] = spib_dma_req[2] GPCFG6[12] = 1 : dma_req[38] = rti2 dma req[1] ,0 : GPCFG6[12] = 0 : dma_req[38] = else spib_dma_req[3] GPCFG6[13] = 1 : dma_req[42] = rti3 dma req[0] ,0 : GPCFG6[13] = 0 : dma_req[42] = else spib_dma_req[4] GPCFG6[14] = 1 : dma_req[43] = rti3 dma req[1] ,0 : GPCFG6[14] = 0 : dma_req[43] = else spib_dma_req[5] GPCFG6[26] = 1 : irq_req[108] = rti2_int_req[0] ,0 : GPCFG6[26] = 0 : irq_req[108] = else epwm2_int1 GPCFG6[27] = 1 : irq_req[109] = rti2_int_req[1] ,0 : GPCFG6[27] = 0 : irq_req[109] = else epwm2_int2 GPCFG6[28] = 1 : irq_req[110] = rti3_int_req[0] ,0 : GPCFG6[28] = 0 : irq_req[110] = else epwm3_int1 GPCFG6[29] = 1 : irq_req[111] = rti3_int_req[1] ,0 : GPCFG6[29] = 0 : irq_req[111] = else epwm3_int2

4.9.3.7 GPCFG11 Register (Offset = 2Ch) [reset = 0h]

GPCFG11 is shown in Figure 4-858 and described in Table 4-899.

Return to Summary Table.

Figure 4-858. GPCFG11 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED						BSS2DSSSWIRQ2	BSS2DSSSWIRQ1
R-0h						0h	0h
15	14	13	12	11	10	9	8
RESERVED						DSS2BSSSWIRQ2	DSS2BSSSWIRQ1
R-0h						0h	0h
7	6	5	4	3	2	1	0
RESERVED						MSS2BSSSWIRQ2	MSS2BSSSWIRQ1
R-0h						0h	0h

Table 4-899. GPCFG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED		0h	Reserved
17	BSS2DSSSWIRQ2		0h	Self clearing register bit to generate interrupt to DSP from BSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
16	BSS2DSSSWIRQ1		0h	Self clearing register bit to generate interrupt to DSP from BSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
15-10	RESERVED	R	0h	Reserved
9	DSS2BSSSWIRQ2		0h	Self clearing register bit to generate interrupt to BSS from DSP. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
8	DSS2BSSSWIRQ1		0h	Self clearing register bit to generate interrupt to BSS from DSP. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
7-2	RESERVED	R	0h	Reserved
1	MSS2BSSSWIRQ2		0h	Self clearing register bit to generate interrupt to BSS from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS
0	MSS2BSSSWIRQ1		0h	Self clearing register bit to generate interrupt to BSS from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to BSS

4.9.3.8 CCCACFG0 Register (Offset = D0h) [reset = 0h]

CCCACFG0 is shown in Figure 4-859 and described in Table 4-900.

Return to Summary Table.

Figure 4-859. CCCACFG0 Register

31	30	29	28	27	26	25	24
CCCA_MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
CCCA_MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU31							CCCA_SINGLE_SHOT_MODE
0h							R/W-0h
7	6	5	4	3	2	1	0
CCCA_ENABLE_MODULE	CCCA_DISABLE_CLOCKS	CCCA_CLOCK1_SEL			CCCA_CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-900. CCCACFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CCCA_MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU31		0h	Reserved
8	CCCA_SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	CCCA_ENABLE_MODULE	R/W	0h	Enable for CCC
6	CCCA_DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CCCA_CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CCCA_CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.9.3.9 CCCACFG1 Register (Offset = D4h) [reset = 0h]

CCCACFG1 is shown in Figure 4-860 and described in Table 4-901.

Return to Summary Table.

Figure 4-860. CCCACFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG1																															
R/W-0h																															

Table 4-901. CCCACFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG1	R/W	0h	count0_expiry_val

4.9.3.10 CCCACFG2 Register (Offset = D8h) [reset = 0h]

CCCACFG2 is shown in Figure 4-861 and described in Table 4-902.

Return to Summary Table.

Figure 4-861. CCCACFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG2																															
R/W-0h																															

Table 4-902. CCCACFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG2	R/W	0h	count1_expected_val

4.9.3.11 CCCACFG3 Register (Offset = DCh) [reset = 0h]

CCCACFG3 is shown in Figure 4-862 and described in Table 4-903.

Return to Summary Table.

Figure 4-862. CCCACFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACFG3																															
R/W-0h																															

Table 4-903. CCCACFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACFG3	R/W	0h	Error Counter value in counter1 clock

4.9.3.12 CCCBCFG0 Register (Offset = E0h) [reset = 0h]

CCCBCFG0 is shown in Figure 4-863 and described in Table 4-904.

Return to Summary Table.

Figure 4-863. CCCBCFG0 Register

31	30	29	28	27	26	25	24
CCCB_MARGIN_COUNT							
R/W-0h							
23	22	21	20	19	18	17	16
CCCB_MARGIN_COUNT							
R/W-0h							
15	14	13	12	11	10	9	8
NU32							CCCB_SINGLE_SHOT_MODE
0h							R/W-0h
7	6	5	4	3	2	1	0
CCCB_ENABLE_MODULE	CCCB_DISABLE_CLOCKS	CCCB_CLOCK1_SEL			CCCB_CLOCK0_SEL		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		

Table 4-904. CCCBCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CCCB_MARGIN_COUNT	R/W	0h	Margin value for clock comparison
15-9	NU32		0h	Reserved
8	CCCB_SINGLE_SHOT_MODE	R/W	0h	1->Single shot mode, 0->Continuous mode
7	CCCB_ENABLE_MODULE	R/W	0h	Enable for CCC
6	CCCB_DISABLE_CLOCKS	R/W	0h	1->Clock cut off, 0-> Normal mode
5-3	CCCB_CLOCK1_SEL	R/W	0h	Slection for Clock 1
2-0	CCCB_CLOCK0_SEL	R/W	0h	Selection for Clock 0

4.9.3.13 CCCBCFG1 Register (Offset = E4h) [reset = 0h]

CCCBCFG1 is shown in Figure 4-864 and described in Table 4-905.

Return to Summary Table.

Figure 4-864. CCCBCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG1																															
R/W-0h																															

Table 4-905. CCCBCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG1	R/W	0h	count0_expiry_val

4.9.3.14 CCCBCFG2 Register (Offset = E8h) [reset = 0h]

CCCBCFG2 is shown in Figure 4-865 and described in Table 4-906.

Return to Summary Table.

Figure 4-865. CCCBCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG2																															
R/W-0h																															

Table 4-906. CCCBCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG2	R/W	0h	count1_expected_val

4.9.3.15 CCCBCFG3 Register (Offset = ECh) [reset = 0h]

CCCBCFG3 is shown in Figure 4-866 and described in Table 4-907.

Return to Summary Table.

Figure 4-866. CCCBCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCFG3																															
R/W-0h																															

Table 4-907. CCCBCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCFG3	R/W	0h	Error Counter value in counter1 clock

4.9.3.16 CCCACNTVAL Register (Offset = F0h) [reset = 0h]

CCCACNTVAL is shown in Figure 4-867 and described in Table 4-908.

Return to Summary Table.

Figure 4-867. CCCACNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCACNTVAL																															
R-0h																															

Table 4-908. CCCACNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCACNTVAL	R	0h	count1_val_out

4.9.3.17 CCCBCNTVAL Register (Offset = F4h) [reset = 0h]

CCCBCNTVAL is shown in Figure 4-868 and described in Table 4-909.

Return to Summary Table.

Figure 4-868. CCCBCNTVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCBCNTVAL																															
R-0h																															

Table 4-909. CCCBCNTVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCBCNTVAL	R	0h	count1_val_out

4.9.3.18 CCCABERRSTAT Register (Offset = F8h) [reset = 0h]

CCCABERRSTAT is shown in Figure 4-869 and described in Table 4-910.

Return to Summary Table.

Figure 4-869. CCCABERRSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCCABERRSTAT																															
R-0h																															

Table 4-910. CCCABERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCCABERRSTAT	R	0h	7:0 : CCCA Error Status 15:8 : CCCB Error Status

4.9.3.19 USERMODEEN Register (Offset = FCh) [reset = 0h]

USERMODEEN is shown in Figure 4-870 and described in Table 4-911.

Return to Summary Table.

Figure 4-870. USERMODEEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERMODEEN																															
R/W-0h																															

Table 4-911. USERMODEEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to MSS GPCFG space.

4.9.3.20 EPWMCFG Register (Offset = 140h) [reset = F00h]

EPWMCFG is shown in Figure 4-871 and described in Table 4-912.

Return to Summary Table.

Figure 4-871. EPWMCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPWMCFG																															
R/W-F00h																															

Table 4-912. EPWMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EPWMCFG	R/W	F00h	[1:0] : 0->rampgen Frame start is SYNCIN to EPWM1 1->FRC Frame Start is SYNCIN to EPWM1 2,3-> External SYNCIN is SYNCIN to EPWM1 [3:2]: 0->rampgen Frame start is SYNCIN to EPWM2 1->FRC Frame Start is SYNCIN to EPWM2 2,3-> EPWM1 SYNCO is SYNCIN to EPWM2 [5:4]: 0->rampgen Frame start is SYNCIN to EPWM3 1->FRC Frame Start is SYNCIN to EPWM3 2,3-> EPWM2 SYNCO is SYNCIN to EPWM3 [31:8] : Reserved

4.9.3.21 DMMSWINT0 Register (Offset = 148h) [reset = 0h]

DMMSWINT0 is shown in Figure 4-872 and described in Table 4-913.

Return to Summary Table.

Figure 4-872. DMMSWINT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINT0																															
0h																															

Table 4-913. DMMSWINT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINT0		0h	HIL Intr0 muxed with frame start interrupt to DSP HIL Intr1 muxed with logical frame start interrupt to DSP HIL Intr2 muxed with Ping/Pong threshold trigger interrupt for DSP HIL Intr3 muxed with ADC data valid interrupt for DSP HIL Intr4 - HIL Intr8 : SW interrupt for DSP HIL Intr9 muxed with frame start interrupt to TPCC0 HIL Intr10 muxed with logical frame start interrupt to TPCC0 HIL Intr11 muxed with Ping/Pong threshold trigger interrupt for TPCC0 HIL Intr12 muxed with ADC data valid interrupt for TPCC0 HIL Intr13 - HIL Intr18 as SW interrupt for TPCC0 HIL Intr19 muxed with frame start interrupt to TPCC1 HIL Intr20 muxed with logical frame start interrupt to TPCC1 HIL Intr21 muxed with Ping/Pong threshold trigger interrupt for TPCC1 HIL Intr22 muxed with ADC data valid interrupt for TPCC1 HIL Intr23 - HIL Intr28 as SW interrupt for TPCC1 HIL Intr29 muxed with frame start interrupt to VIMMR4F HIL Intr30 muxed with logical frame start interrupt to VIMMR4F HIL Intr31 muxed with Ping/Pong threshold trigger interrupt for VIMMR4F HIL Intr32 muxed with ADC data valid interrupt for VIMMR4F HIL Intr33 - HIL Intr38 as SW interrupt for VIMMR4F HIL Intr39 muxed with frame start interrupt right at the source - propagates throughout the system as actual FRC output does. This is muxed with the frame start coming from BSS to TOP so that it propagates to MSS/DSS. HIL Intr40 muxed with frame start interrupt right at the source - propagates throughout the system as actual FRC output does. This is muxed with the ADC clock enable signal coming from FRC. Intr41 muxed with logical frame start interrupt right at the source (FRC) - propagates throughout the system Intr42 muxed with logical frame end interrupt right at the source (FRC) - propagates throughout the system Intr43 muxed with Ping/Pong threshold trigger interrupt right at the source - propagates throughout the system Intr44 muxed with ADC data valid interrupt right at the source - propagates throughout the system

4.9.3.22 DMMSWINT1 Register (Offset = 14Ch) [reset = 0h]

DMMSWINT1 is shown in Figure 4-953 and described in Table 4-996.

Return to Summary Table.

Figure 4-873. DMMSWINT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINT1																															
0h																															

Table 4-914. DMMSWINT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINT1		0h	HIL Interrupts - MSB 32 bits [63:32]

4.9.3.23 DMMSWINTSEL0 Register (Offset = 150h) [reset = 0h]

DMMSWINTSEL0 is shown in Figure 4-874 and described in Table 4-915.

Return to Summary Table.

Figure 4-874. DMMSWINTSEL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINTSEL0																															
R/W-0h																															

Table 4-915. DMMSWINTSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINTSEL0	R/W	0h	Mux control for HIL Interrupts - LSB 32 bits

4.9.3.24 DMMSWINTSEL1 Register (Offset = 154h) [reset = 0h]

DMMSWINTSEL1 is shown in Figure 4-875 and described in Table 4-916.

Return to Summary Table.

Figure 4-875. DMMSWINTSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMMSWINTSEL1																															
R/W-0h																															

Table 4-916. DMMSWINTSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMMSWINTSEL1	R/W	0h	Mux control for HIL Interrupts - MSB 32 bits

4.9.3.25 CCCBWDEN Register (Offset = 158h) [reset = 0h]

CCCBWDEN is shown in Figure 4-876 and described in Table 4-917.

Return to Summary Table.

Figure 4-876. CCCBWDEN Register

31	30	29	28	27	26	25	24
NU36							
R-0h							
23	22	21	20	19	18	17	16
NU36							ENABLECCBERRRSTN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU35							
R-0h							
7	6	5	4	3	2	1	0
NU35							ENABLECCBERRNMI
R-0h							R/W-0h

Table 4-917. CCCBWDEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU36	R	0h	Reserved
16	ENABLECCBERRRSTN	R/W	0h	Enable CCCB error to generate WD restn.In this mode CCCB has to compare MSS CR4 clock to XTAL
15-1	NU35	R	0h	Reserved
0	ENABLECCBERRNMI	R/W	0h	Enable CCCB error to generate NMI.In this mode CCCB has to compare MSS CR4 clock to XTAL

4.9.3.26 GPIOINTREDGESEL Register (Offset = 15Ch) [reset = 0h]

GPIOINTREDGESEL is shown in Figure 4-877 and described in Table 4-918.

Return to Summary Table.

Figure 4-877. GPIOINTREDGESEL Register

31	30	29	28	27	26	25	24
NU37							
0h							
23	22	21	20	19	18	17	16
NU37							
0h							
15	14	13	12	11	10	9	8
NU37							
0h							
7	6	5	4	3	2	1	0
NU37					GPIO2EDGESEL	GPIO1EDGESEL	GPIO0EDGESEL
0h					R/W-0h	R/W-0h	R/W-0h

Table 4-918. GPIOINTREDGESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU37		0h	Reserved
2	GPIO2EDGESEL	R/W	0h	0:Posedge of GPIO2 generates interruptt to MSS CR4 and DSP 1:Negedge
1	GPIO1EDGESEL	R/W	0h	0:Posedge of GPIO1 generates interruptt to MSS CR4 and DSP 1:Negedge
0	GPIO0EDGESEL	R/W	0h	0:Posedge of GPIO0 generates interruptt to MSS CR4 and DSP 1:Negedge

4.9.3.27 PWMDMATRIGEN Register (Offset = 164h) [reset = 0h]

PWMDMATRIGEN is shown in Figure 4-878 and described in Table 4-919.

Return to Summary Table.

Figure 4-878. PWMDMATRIGEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU38															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU38												PWMDMATRIGEN			
R-0h												R/W-0h			

Table 4-919. PWMDMATRIGEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU38	R	0h	Reserved
3-0	PWMDMATRIGEN	R/W	0h	0: "1" -> Mux epwm1a instead of mss_event_gen_1_frc to DMA 1: "1" -> Mux epwm1b instead of mss_event_gen_1_frc to DMA 2: "1" -> Mux epwm2a instead of mss_event_gen_1_frc to DMA 3: "1" -> Mux epwm2b instead of mss_event_gen_1_frc to DMA

4.9.3.28 JTAGTXDATA Register (Offset = 168h) [reset = 0h]

JTAGTXDATA is shown in Figure 4-879 and described in Table 4-920.

Return to Summary Table.

Figure 4-879. JTAGTXDATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGTXDATA																															
R-0h																															

Table 4-920. JTAGTXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGTXDATA	R	0h	Transmit Data. This register is used to pass data to the system security logic

4.9.3.29 JTAGTXCONTROL Register (Offset = 16Ch) [reset = 0h]

JTAGTXCONTROL is shown in Figure 4-880 and described in Table 4-921.

Return to Summary Table.

Figure 4-880. JTAGTXCONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGTXCONTROL																															
R-0h																															

Table 4-921. JTAGTXCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGTXCONTROL	R	0h	Provides the handshake for the JTAGTXDATA Register and can also be used to pass control information to the system security logic. Only bits [31:1] are valid

4.9.3.30 JTAGRXDATA Register (Offset = 170h) [reset = 0h]

JTAGRXDATA is shown in Figure 4-881 and described in Table 4-922.

Return to Summary Table.

Figure 4-881. JTAGRXDATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGRXDATA																															
R/W-0h																															

Table 4-922. JTAGRXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGRXDATA	R/W	0h	Receive data. This register is used to pass data from the system security logic

4.9.3.31 JTAGTXRXACK Register (Offset = 178h) [reset = 0h]

JTAGTXRXACK is shown in Figure 4-882 and described in Table 4-923.

Return to Summary Table.

Figure 4-882. JTAGTXRXACK Register

31	30	29	28	27	26	25	24
NU40							
R-0h							
23	22	21	20	19	18	17	16
NU40							
R-0h							
15	14	13	12	11	10	9	8
NU40							JTAGRXDATAWR
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NU39							JTAGTXDATARD
R-0h							R/W-0h

Table 4-923. JTAGTXRXACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	NU40	R	0h	Reserved
8	JTAGRXDATAWR	R/W	0h	Indication from system security logic that JTAGRXDATA is valid
7-1	NU39	R	0h	Reserved
0	JTAGTXDATARD	R/W	0h	Indication from the system security logic that JTAGTXDATA has been accepted

4.9.3.32 JTAGRXCONTROL Register (Offset = 17Ch) [reset = 0h]

JTAGRXCONTROL is shown in Figure 4-883 and described in Table 4-924.

Return to Summary Table.

Figure 4-883. JTAGRXCONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGRXCONTROL																															
R/W-0h																															

Table 4-924. JTAGRXCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGRXCONTROL	R/W	0h	Provides the handshake for the JTAGRXDATA Register and can also be used to pass control information from the system security logic. Only bits [31:1] is used.

4.9.3.33 MSS2GEMSWIRQ Register (Offset = 180h) [reset = 0h]

MSS2GEMSWIRQ is shown in Figure 4-884 and described in Table 4-925.

Return to Summary Table.

Figure 4-884. MSS2GEMSWIRQ Register

31	30	29	28	27	26	25	24
NU41							
R-0h							
23	22	21	20	19	18	17	16
NU41							
R-0h							
15	14	13	12	11	10	9	8
NU41							
R-0h							
7	6	5	4	3	2	1	0
NU41						MSS2GEMSWIRQ2	MSS2GEMSWIRQ1
R-0h						0h	0h

Table 4-925. MSS2GEMSWIRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU41	R	0h	Reserved
1	MSS2GEMSWIRQ2		0h	Self clearing register bit to generate interrupt to DSP from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP
0	MSS2GEMSWIRQ1		0h	Self clearing register bit to generate interrupt to DSP from MSS. Writing a ‘1’ generates a pulse interrupt on one of the interrupt lines to DSP

4.9.3.34 CSETBFLUSH Register (Offset = 184h) [reset = 0h]

CSETBFLUSH is shown in Figure 4-885 and described in Table 4-926.

Return to Summary Table.

Figure 4-885. CSETBFLUSH Register

31	30	29	28	27	26	25	24
NU43							
R-0h							
23	22	21	20	19	18	17	16
NU43							
R-0h							
15	14	13	12	11	10	9	8
NU43					CSETBFULL	CSETBACQ_OMPLETE	CSETBFLUSHINACK
R-0h					R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
NU42							CSETBFLUSHIN
R-0h							R/W-0h

Table 4-926. CSETBFLUSH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU43	R	0h	Reserved
10	CSETBFULL	R	0h	When HIGH indicates that the ETB RAM has overflowed or wrapped around to address zero
9	CSETBACQ_OMPLETE	R	0h	When HIGH, indicates that trace acquisition is complete by ETB, that is, the trigger counter is at zero
8	CSETBFLUSHINACK	R	0h	Return acknowledgement to CSETBFLUSHIN
7-1	NU42	R	0h	Reserved
0	CSETBFLUSHIN	R/W	0h	External control used to assert the ATB signal AFVALIDS and drain any historical FIFO information on the bus

4.9.4 DSS_REG Registers

Table 4-927 lists the memory-mapped registers for the DSS_REG. All register offset addresses not listed in Table 4-927 should be considered as reserved locations and the register contents should not be modified.

Table 4-927. DSS_REG Registers

Offset	Acronym	Register Name	Section
50h	RTIEVENTCAPTURESEL	RTIEVENTCAPTURESEL	Section 4.9.4.1
6Ch	CQCFG1	CQCFG1	Section 4.9.4.2
80h	TPCCPARSTATCFG	TPCCPARSTATCFG	Section 4.9.4.3
104h	TPTC0WRMPUSTADD0	TPTC0WRMPUSTADD0	Section 4.9.4.4
108h	TPTC0WRMPUSTADD1	TPTC0WRMPUSTADD1	Section 4.9.4.5
10Ch	TPTC0WRMPUSTADD2	TPTC0WRMPUSTADD2	Section 4.9.4.6
110h	TPTC0WRMPUSTADD3	TPTC0WRMPUSTADD3	Section 4.9.4.7
114h	TPTC0WRMPUSTADD4	TPTC0WRMPUSTADD4	Section 4.9.4.8
118h	TPTC0WRMPUSTADD5	TPTC0WRMPUSTADD5	Section 4.9.4.9
124h	TPTC0WRMPUENDADD0	TPTC0WRMPUENDADD0	Section 4.9.4.10
128h	TPTC0WRMPUENDADD1	TPTC0WRMPUENDADD1	Section 4.9.4.11
12Ch	TPTC0WRMPUENDADD2	TPTC0WRMPUENDADD2	Section 4.9.4.12
130h	TPTC0WRMPUENDADD3	TPTC0WRMPUENDADD3	Section 4.9.4.13
134h	TPTC0WRMPUENDADD4	TPTC0WRMPUENDADD4	Section 4.9.4.14
138h	TPTC0WRMPUENDADD5	TPTC0WRMPUENDADD5	Section 4.9.4.15
144h	TPTC0WRMPUERRADD	TPTC0WRMPUERRADD	Section 4.9.4.16
148h	TPTC0RDMPUSTADD0	TPTC0RDMPUSTADD0	Section 4.9.4.17
14Ch	TPTC0RDMPUSTADD1	TPTC0RDMPUSTADD1	Section 4.9.4.18
150h	TPTC0RDMPUSTADD2	TPTC0RDMPUSTADD2	Section 4.9.4.19
154h	TPTC0RDMPUSTADD3	TPTC0RDMPUSTADD3	Section 4.9.4.20
158h	TPTC0RDMPUSTADD4	TPTC0RDMPUSTADD4	Section 4.9.4.21
15Ch	TPTC0RDMPUSTADD5	TPTC0RDMPUSTADD5	Section 4.9.4.22
168h	TPTC0RDMPUENDADD0	TPTC0RDMPUENDADD0	Section 4.9.4.23
16Ch	TPTC0RDMPUENDADD1	TPTC0RDMPUENDADD1	Section 4.9.4.24
170h	TPTC0RDMPUENDADD2	TPTC0RDMPUENDADD2	Section 4.9.4.25
174h	TPTC0RDMPUENDADD3	TPTC0RDMPUENDADD3	Section 4.9.4.26
178h	TPTC0RDMPUENDADD4	TPTC0RDMPUENDADD4	Section 4.9.4.27
17Ch	TPTC0RDMPUENDADD5	TPTC0RDMPUENDADD5	Section 4.9.4.28
188h	TPTC0RDMPUERRADD	TPTC0RDMPUERRADD	Section 4.9.4.29
18Ch	TPTC1WRMPUSTADD0	TPTC1WRMPUSTADD0	Section 4.9.4.30
190h	TPTC1WRMPUSTADD1	TPTC1WRMPUSTADD1	Section 4.9.4.31
194h	TPTC1WRMPUSTADD2	TPTC1WRMPUSTADD2	Section 4.9.4.32
198h	TPTC1WRMPUSTADD3	TPTC1WRMPUSTADD3	Section 4.9.4.33
19Ch	TPTC1WRMPUSTADD4	TPTC1WRMPUSTADD4	Section 4.9.4.34
1A0h	TPTC1WRMPUSTADD5	TPTC1WRMPUSTADD5	Section 4.9.4.35
1ACh	TPTC1WRMPUENDADD0	TPTC1WRMPUENDADD0	Section 4.9.4.36
1B0h	TPTC1WRMPUENDADD1	TPTC1WRMPUENDADD1	Section 4.9.4.37
1B4h	TPTC1WRMPUENDADD2	TPTC1WRMPUENDADD2	Section 4.9.4.38
1B8h	TPTC1WRMPUENDADD3	TPTC1WRMPUENDADD3	Section 4.9.4.39
1BCh	TPTC1WRMPUENDADD4	TPTC1WRMPUENDADD4	Section 4.9.4.40
1C0h	TPTC1WRMPUENDADD5	TPTC1WRMPUENDADD5	Section 4.9.4.41
1CCh	TPTC1WRMPUERRADD	TPTC1WRMPUERRADD	Section 4.9.4.42
1D0h	TPTC1RDMPUSTADD0	TPTC1RDMPUSTADD0	Section 4.9.4.43

Table 4-927. DSS_REG Registers (continued)

Offset	Acronym	Register Name	Section
1D4h	TPTC1RDMPUSTADD1	TPTC1RDMPUSTADD1	Section 4.9.4.44
1D8h	TPTC1RDMPUSTADD2	TPTC1RDMPUSTADD2	Section 4.9.4.45
1DCh	TPTC1RDMPUSTADD3	TPTC1RDMPUSTADD3	Section 4.9.4.46
1E0h	TPTC1RDMPUSTADD4	TPTC1RDMPUSTADD4	Section 4.9.4.47
1E4h	TPTC1RDMPUSTADD5	TPTC1RDMPUSTADD5	Section 4.9.4.48
1F0h	TPTC1RDMPUENDADD0	TPTC1RDMPUENDADD0	Section 4.9.4.49
1F4h	TPTC1RDMPUENDADD1	TPTC1RDMPUENDADD1	Section 4.9.4.50
1F8h	TPTC1RDMPUENDADD2	TPTC1RDMPUENDADD2	Section 4.9.4.51
1FCh	TPTC1RDMPUENDADD3	TPTC1RDMPUENDADD3	Section 4.9.4.52
200h	TPTC1RDMPUENDADD4	TPTC1RDMPUENDADD4	Section 4.9.4.53
204h	TPTC1RDMPUENDADD5	TPTC1RDMPUENDADD5	Section 4.9.4.54
210h	TPTC1RDMPUERRADD	TPTC1RDMPUERRADD	Section 4.9.4.55
214h	TPTCMPUVALIDCFG	TPTCMPUVALIDCFG	Section 4.9.4.56
218h	TPTCMPUENCFG	TPTCMPUENCFG	Section 4.9.4.57
21Ch	TESTPATTERNRX1ICFG	TESTPATTERNRX1ICFG	Section 4.9.4.58
220h	TESTPATTERNRX2ICFG	TESTPATTERNRX2ICFG	Section 4.9.4.59
224h	TESTPATTERNRX3ICFG	TESTPATTERNRX3ICFG	Section 4.9.4.60
228h	TESTPATTERNRX4ICFG	TESTPATTERNRX4ICFG	Section 4.9.4.61
22Ch	TESTPATTERNRX1QCFG	TESTPATTERNRX1QCFG	Section 4.9.4.62
230h	TESTPATTERNRX2QCFG	TESTPATTERNRX2QCFG	Section 4.9.4.63
234h	TESTPATTERNRX3QCFG	TESTPATTERNRX3QCFG	Section 4.9.4.64
238h	TESTPATTERNRX4QCFG	TESTPATTERNRX4QCFG	Section 4.9.4.65
23Ch	TESTPATTERNVLDCFG	TESTPATTERNVLDCFG	Section 4.9.4.66
240h	DSSMISC	DSSMISC	
258h	TPCC1PARSTATCFG	TPCC1PARSTATCFG	Section 4.9.4.67
260h	DMMSWINT1	DMMSWINT1	Section 4.9.4.68
270h	DSSINTRCFG	DSSINTRCFG	Section 4.9.4.69
274h	MPUMSTIDCFG1	MPUMSTIDCFG1	Section 4.9.4.70
278h	MPUMSTIDCFG2	MPUMSTIDCFG2	Section 4.9.4.71
27Ch	MPUMSTIDCFG3	MPUMSTIDCFG3	Section 4.9.4.72
280h	HSRAM1ECCCFG	HSRAM1ECCCFG	Section 4.9.4.73
288h	DATATRRAMECCCFG	DATATRRAMECCCFG	Section 4.9.4.74
28Ch	ADCBUFPINGECCCFG	ADCBUFPINGECCCFG	Section 4.9.4.75
290h	ADCBUFPONGECCCFG	ADCBUFPONGECCCFG	Section 4.9.4.76
29Ch	UMAP0PARITYCFG1	UMAP0PARITYCFG1	Section 4.9.4.77
2A0h	UMAP0PARITYCFG2	UMAP0PARITYCFG2	Section 4.9.4.78
2A4h	UMAP0PARITYCFG3	UMAP0PARITYCFG3	Section 4.9.4.79
2A8h	UMAP1PARITYCFG1	UMAP1PARITYCFG1	Section 4.9.4.80
2ACh	UMAP1PARITYCFG2	UMAP1PARITYCFG2	Section 4.9.4.81
2B0h	UMAP1PARITYCFG3	UMAP1PARITYCFG3	Section 4.9.4.82
2B4h	ESMGRP2MASKCFG	ESMGRP2MASKCFG	Section 4.9.4.83
2B8h	L2MEMINITCFG1[(1)]	L2MEMINITCFG1	Section 4.9.4.84
2BCh	L2MEMINITCFG2[(1)]	L2MEMINITCFG2	Section 4.9.4.85
2C0h	GEMRSTCAUSE[(1)]	GEMRSTCAUSE	Section 4.9.4.86
2CCh	GEMPWRSMCFG4[(1)]	GEMPWRSMCFG4	Section 4.9.4.87
2D4h	PWRSMWAKEMASK0[(1)]	PWRSMWAKEMASK0	Section 4.9.4.88

(1) Applicable for devices with DSP C674x (such as the 68xx device).

Table 4-927. DSS_REG Registers (continued)

Offset	Acronym	Register Name	Section
2D8h	PWRSMWAKEMASK1[(1)]	PWRSMWAKEMASK1	Section 4.9.4.89
2DCh	PWRSMWAKEMASK2[(1)]	PWRSMWAKEMASK2	Section 4.9.4.90
2E0h	PWRSMMISEVTMASK0[(1)]	PWRSMMISEVTMASK0	Section 4.9.4.91
2E4h	PWRSMMISEVTMASK1[(1)]	PWRSMMISEVTMASK1	Section 4.9.4.92
2E8h	PWRSMMISEVTMASK2[(1)]	PWRSMMISEVTMASK2	Section 4.9.4.93
2ECh	PWRSMWAKESRCSTAT0[(1)]	PWRSMWAKESRCSTAT0	Section 4.9.4.94
2F0h	PWRSMWAKESRCSTAT1[(1)]	PWRSMWAKESRCSTAT1	Section 4.9.4.95
320h	PWRSMWAKESRCSTAT2[(1)]	PWRSMWAKESRCSTAT2	Section 4.9.4.96
324h	PWRSMEVNTMONSTAT0[(1)]	PWRSMEVNTMONSTAT0	Section 4.9.4.97
328h	PWRSMEVNTMONSTAT1[(1)]	PWRSMEVNTMONSTAT1	Section 4.9.4.98
32Ch	PWRSMEVNTMONSTAT2[(1)]	PWRSMEVNTMONSTAT2	Section 4.9.4.99
330h	PWRSMWAKESRCSTATCLR0[(1)]	PWRSMWAKESRCSTATCLR0	Section 4.9.4.100
334h	PWRSMWAKESRCSTATCLR1[(1)]	PWRSMWAKESRCSTATCLR1	Section 4.9.4.101
338h	PWRSMWAKESRCSTATCLR2[(1)]	PWRSMWAKESRCSTATCLR2	Section 4.9.4.102
33Ch	ADCBUFCFG1	ADCBUFCFG1	Section 4.9.4.103
340h	ADCBUFCFG2	ADCBUFCFG2	Section 4.9.4.104
344h	ADCBUFCFG3	ADCBUFCFG3	Section 4.9.4.105
348h	ADCBUFCFG4	ADCBUFCFG4	Section 4.9.4.106
34Ch	STCPBISTSMCFG1	STCPBISTSMCFG1	Section 4.9.4.107
350h	STCPBISTSMCFG2	STCPBISTSMCFG2	Section 4.9.4.108
358h	RTI2EVENTCAPTURESEL	RTI2EVENTCAPTURESEL	Section 4.9.4.109
35Ch	DSSMISC5	DSSMISC5	Section 4.9.4.110

Complex bit access types are encoded to fit into small table cells. Table 4-928 shows the codes that are used for access types in this section.

Table 4-928. DSS_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.9.4.1 RTIEVENTCAPTURESEL Register (Offset = 50h) [reset = 0h]

RTIEVENTCAPTURESEL is shown in Figure 4-886 and described in Table 4-929.

Return to Summary Table.

Figure 4-886. RTIEVENTCAPTURESEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2									EVT1							NU1									EVT0							
R-0h									R/W-0h							R-0h									R/W-0h						

Table 4-929. RTIEVENTCAPTURESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	
22-16	EVT1	R/W	0h	Setting the source of interrupt for Counter value capture for RTI1 Event1
15-7	NU1	R	0h	
6-0	EVT0	R/W	0h	Setting the source of interrupt for Counter value capture for RTI1 Event0.

4.9.4.2 CQCFG1 Register (Offset = 6Ch) [reset = 40100000h]

CQCFG1 is shown in Figure 4-887 and described in Table 4-930.

Return to Summary Table.

Figure 4-887. CQCFG1 Register

31	30	29	28	27	26	25	24
NU3	CQ2BASEADDR						
R-0h	R/W-100h						
23	22	21	20	19	18	17	16
CQ2BASEADDR			CQ1BASEADDR				
R/W-100h			R/W-80h				
15	14	13	12	11	10	9	8
CQ1BASEADDR			CQ0BASEADDR				
R/W-80h			R/W-0h				
7	6	5	4	3	2	1	0
CQ0BASEADDR				CQ96BITPACKEN	NU	CQDATAWIDTH	
R/W-0h				R/W-0h	R-0h	R/W-0h	

Table 4-930. CQCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU3	R	0h	
30-22	CQ2BASEADDR	R/W	100h	128-bit Address offset which inidcates the start address for storing CQ0 (ADC/RxIF Saturation Detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
21-13	CQ1BASEADDR	R/W	80h	128-bit Address offset which inidcates the start address for storing CQ0 (Signal Image Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
12-4	CQ0BASEADDR	R/W	0h	128-bit Address offset which inidcates the start address for storing CQ0 (Wide Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
3	CQ96BITPACKEN	R/W	0h	This is used to pack the CQ data into only the LSB 96 bits of each row of the CQ memory. This can be used in 3 channel mode of LVDS where the ADC data and Chirp Params occupy only LSB 96 bits of each memory row.
2	NU	R	0h	
1-0	CQDATAWIDTH	R/W	0h	This is used to appropriately pack the valid CQ data bits in appropriate bits in the CQ memory. 00, 01->Raw 16, 10-->Raw 12, 11-->Raw14

4.9.4.3 TPCCPARSTATCFG Register (Offset = 80h) [reset = 0h]

TPCCPARSTATCFG is shown in Figure 4-888 and described in Table 4-931.

Return to Summary Table.

Figure 4-888. TPCCPARSTATCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU					TPCCPARITYTSTEN	TPCCPARITYEN	TPCCPARITYCLR
R-0h					R/W-0h	R/W-0h	0h
7	6	5	4	3	2	1	0
TPCCPARITYSTAT							
R-0h							

Table 4-931. TPCCPARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	
10	TPCCPARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in TPCC
9	TPCCPARITYEN	R/W	0h	Enable bit for the Parity computation in TPCC
8	TPCCPARITYCLR		0h	Write 0x1 to clear the status. This is a special access type; a write to this field generates a pulse.
7-0	TPCCPARITYSTAT	R	0h	Parity address from TPCC

4.9.4.4 TPTC0WRMPUSTADD0 Register (Offset = 104h) [reset = 0h]

TPTC0WRMPUSTADD0 is shown in Figure 4-889 and described in Table 4-932.

Return to Summary Table.

Figure 4-889. TPTC0WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD0																															
R/W-0h																															

Table 4-932. TPTC0WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC0

4.9.4.5 TPTC0WRMPUSTADD1 Register (Offset = 108h) [reset = 0h]

TPTC0WRMPUSTADD1 is shown in Figure 4-890 and described in Table 4-933.

Return to Summary Table.

Figure 4-890. TPTC0WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD1																															
R/W-0h																															

Table 4-933. TPTC0WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC0

4.9.4.6 TPTC0WRMPUSTADD2 Register (Offset = 10Ch) [reset = 0h]

TPTC0WRMPUSTADD2 is shown in Figure 4-891 and described in Table 4-934.

Return to Summary Table.

Figure 4-891. TPTC0WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD2																															
R/W-0h																															

Table 4-934. TPTC0WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC0

4.9.4.7 TPTC0WRMPUSTADD3 Register (Offset = 110h) [reset = 0h]

TPTC0WRMPUSTADD3 is shown in Figure 4-892 and described in Table 4-935.

Return to Summary Table.

Figure 4-892. TPTC0WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD3																															
R/W-0h																															

Table 4-935. TPTC0WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC0

4.9.4.8 TPTC0WRMPUSTADD4 Register (Offset = 114h) [reset = 0h]

TPTC0WRMPUSTADD4 is shown in Figure 4-893 and described in Table 4-936.

Return to Summary Table.

Figure 4-893. TPTC0WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD4																															
R/W-0h																															

Table 4-936. TPTC0WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC0

4.9.4.9 TPTC0WRMPUSTADD5 Register (Offset = 118h) [reset = 0h]

TPTC0WRMPUSTADD5 is shown in Figure 4-894 and described in Table 4-937.

Return to Summary Table.

Figure 4-894. TPTC0WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUSTADD5																															
R/W-0h																															

Table 4-937. TPTC0WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC0

4.9.4.10 TPTC0WRMPUENDADD0 Register (Offset = 124h) [reset = 0h]

TPTC0WRMPUENDADD0 is shown in Figure 4-895 and described in Table 4-938.

Return to Summary Table.

Figure 4-895. TPTC0WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD0																															
R/W-0h																															

Table 4-938. TPTC0WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC0

4.9.4.11 TPTC0WRMPUENDADD1 Register (Offset = 128h) [reset = 0h]

TPTC0WRMPUENDADD1 is shown in Figure 4-896 and described in Table 4-939.

Return to Summary Table.

Figure 4-896. TPTC0WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD1																															
R/W-0h																															

Table 4-939. TPTC0WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC0

4.9.4.12 TPTC0WRMPUENDADD2 Register (Offset = 12Ch) [reset = 0h]

TPTC0WRMPUENDADD2 is shown in Figure 4-897 and described in Table 4-940.

Return to Summary Table.

Figure 4-897. TPTC0WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD2																															
R/W-0h																															

Table 4-940. TPTC0WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC0

4.9.4.13 TPTC0WRMPUENDADD3 Register (Offset = 130h) [reset = 0h]

TPTC0WRMPUENDADD3 is shown in Figure 4-898 and described in Table 4-941.

Return to Summary Table.

Figure 4-898. TPTC0WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD3																															
R/W-0h																															

Table 4-941. TPTC0WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC0

4.9.4.14 TPTC0WRMPUENDADD4 Register (Offset = 134h) [reset = 0h]

TPTC0WRMPUENDADD4 is shown in Figure 4-899 and described in Table 4-942.

Return to Summary Table.

Figure 4-899. TPTC0WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD4																															
R/W-0h																															

Table 4-942. TPTC0WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC0

4.9.4.15 TPTC0WRMPUENDADD5 Register (Offset = 138h) [reset = 0h]

TPTC0WRMPUENDADD5 is shown in Figure 4-900 and described in Table 4-943.

Return to Summary Table.

Figure 4-900. TPTC0WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUENDADD5																															
R/W-0h																															

Table 4-943. TPTC0WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC0

4.9.4.16 TPTC0WRMPUERRADD Register (Offset = 144h) [reset = 0h]

TPTC0WRMPUERRADD is shown in Figure 4-901 and described in Table 4-944.

Return to Summary Table.

Figure 4-901. TPTC0WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0WRMPUERRADD																															
R-0h																															

Table 4-944. TPTC0WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC0

4.9.4.17 TPTC0RDMPUSTADD0 Register (Offset = 148h) [reset = 0h]

TPTC0RDMPUSTADD0 is shown in Figure 4-902 and described in Table 4-945.

Return to Summary Table.

Figure 4-902. TPTC0RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD0																															
R/W-0h																															

Table 4-945. TPTC0RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC0

4.9.4.18 TPTC0RDMPUSTADD1 Register (Offset = 14Ch) [reset = 0h]

TPTC0RDMPUSTADD1 is shown in Figure 4-903 and described in Table 4-946.

Return to Summary Table.

Figure 4-903. TPTC0RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD1																															
R/W-0h																															

Table 4-946. TPTC0RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC0

4.9.4.19 TPTC0RDMPUSTADD2 Register (Offset = 150h) [reset = 0h]

TPTC0RDMPUSTADD2 is shown in Figure 4-904 and described in Table 4-947.

Return to Summary Table.

Figure 4-904. TPTC0RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD2																															
R/W-0h																															

Table 4-947. TPTC0RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC0

4.9.4.20 TPTC0RDMPUSTADD3 Register (Offset = 154h) [reset = 0h]

TPTC0RDMPUSTADD3 is shown in Figure 4-905 and described in Table 4-948.

Return to Summary Table.

Figure 4-905. TPTC0RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD3																															
R/W-0h																															

Table 4-948. TPTC0RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC0

4.9.4.21 TPTC0RDMPUSTADD4 Register (Offset = 158h) [reset = 0h]

TPTC0RDMPUSTADD4 is shown in Figure 4-906 and described in Table 4-949.

Return to Summary Table.

Figure 4-906. TPTC0RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD4																															
R/W-0h																															

Table 4-949. TPTC0RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC0

4.9.4.22 TPTC0RDMPUSTADD5 Register (Offset = 15Ch) [reset = 0h]

TPTC0RDMPUSTADD5 is shown in Figure 4-907 and described in Table 4-950.

Return to Summary Table.

Figure 4-907. TPTC0RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUSTADD5																															
R/W-0h																															

Table 4-950. TPTC0RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC0

4.9.4.23 TPTC0RDMPUENDADD0 Register (Offset = 168h) [reset = 0h]

TPTC0RDMPUENDADD0 is shown in Figure 4-908 and described in Table 4-951.

Return to Summary Table.

Figure 4-908. TPTC0RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD0																															
R/W-0h																															

Table 4-951. TPTC0RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC0

4.9.4.24 TPTC0RDMPUENDADD1 Register (Offset = 16Ch) [reset = 0h]

TPTC0RDMPUENDADD1 is shown in Figure 4-909 and described in Table 4-952.

Return to Summary Table.

Figure 4-909. TPTC0RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD1																															
R/W-0h																															

Table 4-952. TPTC0RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC0

4.9.4.25 TPTC0RDMPUENDADD2 Register (Offset = 170h) [reset = 0h]

TPTC0RDMPUENDADD2 is shown in Figure 4-910 and described in Table 4-953.

Return to Summary Table.

Figure 4-910. TPTC0RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD2																															
R/W-0h																															

Table 4-953. TPTC0RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC0

4.9.4.26 TPTC0RDMPUENDADD3 Register (Offset = 174h) [reset = 0h]

TPTC0RDMPUENDADD3 is shown in Figure 4-911 and described in Table 4-954.

Return to Summary Table.

Figure 4-911. TPTC0RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD3																															
R/W-0h																															

Table 4-954. TPTC0RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC0

4.9.4.27 TPTC0RDMPUENDADD4 Register (Offset = 178h) [reset = 0h]

TPTC0RDMPUENDADD4 is shown in Figure 4-912 and described in Table 4-955.

Return to Summary Table.

Figure 4-912. TPTC0RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD4																															
R/W-0h																															

Table 4-955. TPTC0RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC0

4.9.4.28 TPTC0RDMPUENDADD5 Register (Offset = 17Ch) [reset = 0h]

TPTC0RDMPUENDADD5 is shown in Figure 4-913 and described in Table 4-956.

Return to Summary Table.

Figure 4-913. TPTC0RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUENDADD5																															
R/W-0h																															

Table 4-956. TPTC0RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC0

4.9.4.29 TPTC0RDMPUERRADD Register (Offset = 188h) [reset = 0h]

TPTC0RDMPUERRADD is shown in Figure 4-914 and described in Table 4-957.

Return to Summary Table.

Figure 4-914. TPTC0RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPUERRADD																															
R-0h																															

Table 4-957. TPTC0RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC0RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC0

4.9.4.30 TPTC1WRMPUSTADD0 Register (Offset = 18Ch) [reset = 0h]

TPTC1WRMPUSTADD0 is shown in Figure 4-915 and described in Table 4-958.

Return to Summary Table.

Figure 4-915. TPTC1WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD0																															
R/W-0h																															

Table 4-958. TPTC1WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC1

4.9.4.31 TPTC1WRMPUSTADD1 Register (Offset = 190h) [reset = 0h]

TPTC1WRMPUSTADD1 is shown in Figure 4-916 and described in Table 4-959.

Return to Summary Table.

Figure 4-916. TPTC1WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD1																															
R/W-0h																															

Table 4-959. TPTC1WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC1

4.9.4.32 TPTC1WRMPUSTADD2 Register (Offset = 194h) [reset = 0h]

TPTC1WRMPUSTADD2 is shown in Figure 4-917 and described in Table 4-960.

Return to Summary Table.

Figure 4-917. TPTC1WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD2																															
R/W-0h																															

Table 4-960. TPTC1WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC1

4.9.4.33 TPTC1WRMPUSTADD3 Register (Offset = 198h) [reset = 0h]

TPTC1WRMPUSTADD3 is shown in Figure 4-918 and described in Table 4-961.

Return to Summary Table.

Figure 4-918. TPTC1WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD3																															
R/W-0h																															

Table 4-961. TPTC1WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC1

4.9.4.34 TPTC1WRMPUSTADD4 Register (Offset = 19Ch) [reset = 0h]

TPTC1WRMPUSTADD4 is shown in Figure 4-919 and described in Table 4-962.

Return to Summary Table.

Figure 4-919. TPTC1WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD4																															
R/W-0h																															

Table 4-962. TPTC1WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC1

4.9.4.35 TPTC1WRMPUSTADD5 Register (Offset = 1A0h) [reset = 0h]

TPTC1WRMPUSTADD5 is shown in Figure 4-920 and described in Table 4-963.

Return to Summary Table.

Figure 4-920. TPTC1WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUSTADD5																															
R/W-0h																															

Table 4-963. TPTC1WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC1

4.9.4.36 TPTC1WRMPUENDADD0 Register (Offset = 1ACh) [reset = 0h]

TPTC1WRMPUENDADD0 is shown in Figure 4-921 and described in Table 4-964.

Return to Summary Table.

Figure 4-921. TPTC1WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD0																															
R/W-0h																															

Table 4-964. TPTC1WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC1

4.9.4.37 TPTC1WRMPUENDADD1 Register (Offset = 1B0h) [reset = 0h]

TPTC1WRMPUENDADD1 is shown in Figure 4-922 and described in Table 4-965.

Return to Summary Table.

Figure 4-922. TPTC1WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD1																															
R/W-0h																															

Table 4-965. TPTC1WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC1

4.9.4.38 TPTC1WRMPUENDADD2 Register (Offset = 1B4h) [reset = 0h]

TPTC1WRMPUENDADD2 is shown in Figure 4-923 and described in Table 4-966.

Return to Summary Table.

Figure 4-923. TPTC1WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD2																															
R/W-0h																															

Table 4-966. TPTC1WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC1

4.9.4.39 TPTC1WRMPUENDADD3 Register (Offset = 1B8h) [reset = 0h]

TPTC1WRMPUENDADD3 is shown in Figure 4-924 and described in Table 4-967.

Return to Summary Table.

Figure 4-924. TPTC1WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD3																															
R/W-0h																															

Table 4-967. TPTC1WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC1

4.9.4.40 TPTC1WRMPUENDADD4 Register (Offset = 1BCh) [reset = 0h]

TPTC1WRMPUENDADD4 is shown in Figure 4-925 and described in Table 4-968.

Return to Summary Table.

Figure 4-925. TPTC1WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD4																															
R/W-0h																															

Table 4-968. TPTC1WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC1

4.9.4.41 TPTC1WRMPUENDADD5 Register (Offset = 1C0h) [reset = 0h]

TPTC1WRMPUENDADD5 is shown in Figure 4-926 and described in Table 4-969.

Return to Summary Table.

Figure 4-926. TPTC1WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUENDADD5																															
R/W-0h																															

Table 4-969. TPTC1WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC1

4.9.4.42 TPTC1WRMPUERRADD Register (Offset = 1CCh) [reset = 0h]

TPTC1WRMPUERRADD is shown in Figure 4-927 and described in Table 4-970.

Return to Summary Table.

Figure 4-927. TPTC1WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1WRMPUERRADD																															
R-0h																															

Table 4-970. TPTC1WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC1

4.9.4.43 TPTC1RDMPUSTADD0 Register (Offset = 1D0h) [reset = 0h]

TPTC1RDMPUSTADD0 is shown in Figure 4-928 and described in Table 4-971.

Return to Summary Table.

Figure 4-928. TPTC1RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD0																															
R/W-0h																															

Table 4-971. TPTC1RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC1

4.9.4.44 TPTC1RDMPUSTADD1 Register (Offset = 1D4h) [reset = 0h]

TPTC1RDMPUSTADD1 is shown in Figure 4-929 and described in Table 4-972.

Return to Summary Table.

Figure 4-929. TPTC1RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD1																															
R/W-0h																															

Table 4-972. TPTC1RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC1

4.9.4.45 TPTC1RDMPUSTADD2 Register (Offset = 1D8h) [reset = 0h]

TPTC1RDMPUSTADD2 is shown in Figure 4-930 and described in Table 4-973.

Return to Summary Table.

Figure 4-930. TPTC1RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD2																															
R/W-0h																															

Table 4-973. TPTC1RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC1

4.9.4.46 TPTC1RDMPUSTADD3 Register (Offset = 1DCh) [reset = 0h]

TPTC1RDMPUSTADD3 is shown in Figure 4-931 and described in Table 4-974.

Return to Summary Table.

Figure 4-931. TPTC1RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD3																															
R/W-0h																															

Table 4-974. TPTC1RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC1

4.9.4.47 TPTC1RDMPUSTADD4 Register (Offset = 1E0h) [reset = 0h]

TPTC1RDMPUSTADD4 is shown in Figure 4-932 and described in Table 4-975.

Return to Summary Table.

Figure 4-932. TPTC1RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD4																															
R/W-0h																															

Table 4-975. TPTC1RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC1

4.9.4.48 TPTC1RDMPUSTADD5 Register (Offset = 1E4h) [reset = 0h]

TPTC1RDMPUSTADD5 is shown in Figure 4-933 and described in Table 4-976.

Return to Summary Table.

Figure 4-933. TPTC1RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUSTADD5																															
R/W-0h																															

Table 4-976. TPTC1RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC1

4.9.4.49 TPTC1RDMPUENDADD0 Register (Offset = 1F0h) [reset = 0h]

TPTC1RDMPUENDADD0 is shown in Figure 4-934 and described in Table 4-977.

Return to Summary Table.

Figure 4-934. TPTC1RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD0																															
R/W-0h																															

Table 4-977. TPTC1RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC1

4.9.4.50 TPTC1RDMPUENDADD1 Register (Offset = 1F4h) [reset = 0h]

TPTC1RDMPUENDADD1 is shown in Figure 4-935 and described in Table 4-978.

Return to Summary Table.

Figure 4-935. TPTC1RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD1																															
R/W-0h																															

Table 4-978. TPTC1RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC1

4.9.4.51 TPTC1RDMPUENDADD2 Register (Offset = 1F8h) [reset = 0h]

TPTC1RDMPUENDADD2 is shown in Figure 4-936 and described in Table 4-979.

Return to Summary Table.

Figure 4-936. TPTC1RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD2																															
R/W-0h																															

Table 4-979. TPTC1RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC1

4.9.4.52 TPTC1RDMPUENDADD3 Register (Offset = 1FCh) [reset = 0h]

TPTC1RDMPUENDADD3 is shown in Figure 4-937 and described in Table 4-980.

Return to Summary Table.

Figure 4-937. TPTC1RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD3																															
R/W-0h																															

Table 4-980. TPTC1RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC1

4.9.4.53 TPTC1RDMPUENDADD4 Register (Offset = 200h) [reset = 0h]

TPTC1RDMPUENDADD4 is shown in Figure 4-938 and described in Table 4-981.

Return to Summary Table.

Figure 4-938. TPTC1RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD4																															
R/W-0h																															

Table 4-981. TPTC1RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC1

4.9.4.54 TPTC1RDMPUENDADD5 Register (Offset = 204h) [reset = 0h]

TPTC1RDMPUENDADD5 is shown in Figure 4-939 and described in Table 4-982.

Return to Summary Table.

Figure 4-939. TPTC1RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUENDADD5																															
R/W-0h																															

Table 4-982. TPTC1RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC1

4.9.4.55 TPTC1RDMPUERRADD Register (Offset = 210h) [reset = 0h]

TPTC1RDMPUERRADD is shown in Figure 4-940 and described in Table 4-983.

Return to Summary Table.

Figure 4-940. TPTC1RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC1RDMPUERRADD																															
R-0h																															

Table 4-983. TPTC1RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC1RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC1

4.9.4.56 TPTCMPUVALIDCFG Register (Offset = 214h) [reset = 0h]

TPTCMPUVALIDCFG is shown in Figure 4-941 and described in Table 4-984.

Return to Summary Table.

Figure 4-941. TPTCMPUVALIDCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPTC1RDMPURNGVLD								TPTC1WRMPURNGVLD							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC0RDMPURNGVLD								TPTC0WRMPURNGVLD							
R/W-0h								R/W-0h							

Table 4-984. TPTCMPUVALIDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TPTC1RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC1. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
23-16	TPTC1WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC1. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
15-8	TPTC0RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC0. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
7-0	TPTC0WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC0. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled

4.9.4.57 TPTCMPUENCFG Register (Offset = 218h) [reset = 0h]

TPTCMPUENCFG is shown in Figure 4-942 and described in Table 4-985.

Return to Summary Table.

Figure 4-942. TPTCMPUENCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPTC1RDMPUERRCLR	TPTC1WRMPUERRCLR	TPTC0RDMPUERRCLR	TPTC0WRMPUERRCLR	TPTC1RDMPUEN	TPTC1WRMPUEN	TPTC0RDMPUEN	TPTC0WRMPUEN
0h	0h	0h	0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-985. TPTCMPUENCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPTC1RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC1. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
6	TPTC1WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC1. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
5	TPTC0RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC0. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
4	TPTC0WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC0. Write 0x1 to clear the MPU error: it is a wspecial access type - a write to this field generate a pulse.
3	TPTC1RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC1. 0 : MPU is disabled 1 : MPU is enabled
2	TPTC1WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC1. 0 : MPU is disabled 1 : MPU is enabled
1	TPTC0RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC0. 0 : MPU is disabled 1 : MPU is enabled
0	TPTC0WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC0. 0 : MPU is disabled 1 : MPU is enabled

4.9.4.58 TESTPATTERNRX1ICFG Register (Offset = 21Ch) [reset = 00010000h]

TESTPATTERNRX1ICFG is shown in Figure 4-943 and described in Table 4-986.

Return to Summary Table.

Figure 4-943. TESTPATTERNRX1ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1IINCR																TSTPATRX1IOFFSET															
R/W-1h																R/W-0h															

Table 4-986. TESTPATTERNRX1ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.9.4.59 TESTPATTERNRX2ICFG Register (Offset = 220h) [reset = 00010000h]

TESTPATTERNRX2ICFG is shown in Figure 4-944 and described in Table 4-987.

Return to Summary Table.

Figure 4-944. TESTPATTERNRX2ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2IINCR																TSTPATRX2IOFFSET															
R/W-1h																R/W-0h															

Table 4-987. TESTPATTERNRX2ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 1.
15-0	TSTPATRX2IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 1.

4.9.4.60 TESTPATTERNRX3ICFG Register (Offset = 224h) [reset = 00010000h]

TESTPATTERNRX3ICFG is shown in Figure 4-945 and described in Table 4-988.

Return to Summary Table.

Figure 4-945. TESTPATTERNRX3ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3IINCR																TSTPATRX3IOFFSET															
R/W-1h																R/W-0h															

Table 4-988. TESTPATTERNRX3ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 2
15-0	TSTPATRX3IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 2

4.9.4.61 TESTPATTERNRX4ICFG Register (Offset = 228h) [reset = 00010000h]

TESTPATTERNRX4ICFG is shown in Figure 4-946 and described in Table 4-989.

Return to Summary Table.

Figure 4-946. TESTPATTERNRX4ICFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4IINCR																TSTPATRX4IOFFSET															
R/W-1h																R/W-0h															

Table 4-989. TESTPATTERNRX4ICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 3
15-0	TSTPATRX4IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 3

4.9.4.62 TESTPATTERNRX1QCFG Register (Offset = 22Ch) [reset = 00010000h]

TESTPATTERNRX1QCFG is shown in Figure 4-947 and described in Table 4-990.

Return to Summary Table.

Figure 4-947. TESTPATTERNRX1QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX1QINCR																TSTPATRX1QOFFSET															
R/W-1h																R/W-0h															

Table 4-990. TESTPATTERNRX1QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

4.9.4.63 TESTPATTERNRX2QCFG Register (Offset = 230h) [reset = 00010000h]

TESTPATTERNRX2QCFG is shown in Figure 4-948 and described in Table 4-991.

Return to Summary Table.

Figure 4-948. TESTPATTERNRX2QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX2QINCR																TSTPATRX2QOFFSET															
R/W-1h																R/W-0h															

Table 4-991. TESTPATTERNRX2QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 1.
15-0	TSTPATRX2QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 1.

4.9.4.64 TESTPATTERNRX3QCFG Register (Offset = 234h) [reset = 00010000h]

TESTPATTERNRX3QCFG is shown in Figure 4-949 and described in Table 4-992.

Return to Summary Table.

Figure 4-949. TESTPATTERNRX3QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX3QINCR																TSTPATRX3QOFFSET															
R/W-1h																R/W-0h															

Table 4-992. TESTPATTERNRX3QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 2
15-0	TSTPATRX3QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 2

4.9.4.65 TESTPATTERNRX4QCFG Register (Offset = 238h) [reset = 00010000h]

TESTPATTERNRX4QCFG is shown in Figure 4-950 and described in Table 4-993.

Return to Summary Table.

Figure 4-950. TESTPATTERNRX4QCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTPATRX4QINCR																TSTPATRX4QOFFSET															
R/W-1h																R/W-0h															

Table 4-993. TESTPATTERNRX4QCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 3
15-0	TSTPATRX4QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 3

4.9.4.66 TESTPATTERNVLDCFG Register (Offset = 23Ch) [reset = 8h]

TESTPATTERNVLDCFG is shown in Figure 4-951 and described in Table 4-994.

Return to Summary Table.

Figure 4-951. TESTPATTERNVLDCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU					TSTPATGENEN		
R-0h					R/W-0h		
7	6	5	4	3	2	1	0
TSTPATVLDCNT							
R/W-8h							

Table 4-994. TESTPATTERNVLDCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	
10-8	TSTPATGENEN	R/W	0h	Enable for test pattern generator. This is used to Mux with the functional data from BSS. 000 -->Disable, 111-->Enable, Others are reserved.
7-0	TSTPATVLDCNT	R/W	8h	Number of DSS Interconnect clocks (200 MHz) between successive samples for the test pattern gen.

4.9.4.67 TPCC1PARSTATCFG Register (Offset = 258h) [reset = 0h]

TPCC1PARSTATCFG is shown in Figure 4-952 and described in Table 4-995.

Return to Summary Table.

Figure 4-952. TPCC1PARSTATCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU				TPCC1PARITYTSTEN	TPCC1PARITYEN	TPCC1PARITYCLR	TPCC1PARITYSTAT
R-0h				R/W-0h	R/W-0h	0h	R-0h
7	6	5	4	3	2	1	0
TPCC1PARITYSTAT							
R-0h							

Table 4-995. TPCC1PARSTATCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11	TPCC1PARITYTSTEN	R/W	0h	Enable bit for the self test of the Parity logic in TPCC
10	TPCC1PARITYEN	R/W	0h	Enable bit for the Parity computation in TPCC
9	TPCC1PARITYCLR		0h	Clear bit for the Parity error from TPCC Write 0x1 to clear the status. It is a wspecial access type; a write to this field generates a pulse.
8-0	TPCC1PARITYSTAT	R	0h	Parity address from TPCC

4.9.4.68 DMMSWINT1 Register (Offset = 260h) [reset = 0h]

DMMSWINT1 is shown in Figure 4-953 and described in Table 4-996.

Return to Summary Table.

Figure 4-953. DMMSWINT1 Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2	DMMCQWREN	DMMCQPINPONSEL	DMMCPBPMMEMSEL	DMMCPBPMWREN	DMMCPBPMPINPONSEL	DMMADCBUFWREN	DMMADCBUFPINPONSEL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
NU1							
R-0h							
7	6	5	4	3	2	1	0
NU1							
R-0h							

Table 4-996. DMMSWINT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	Not Used
22	DMMCQWREN	R/W	0h	CQ Write Enable from DMM. 0 --> Write to CQ memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from CQ_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMCQPINPONSEL register.
21	DMMCQPINPONSEL	R/W	0h	CQ Ping Pong select for HIL Mode
20	DMMCPBPMMEMSEL	R/W	0h	Select signal for Muxing between HW Registers/Memory for CPBPM data. 0 --> Read access from CPBPM_MEM slave in DSS interconnect will be routed to HW Registers which is populated by DFE/RampGen, 1 --> Read access from CPBPM_MEM slave in DSS interconnect will be routed to appropriate CPBPM memory (Ping/Pong).
19	DMMCPBPMWREN	R/W	0h	CPBPM Write Enable from DMM. 0 --> Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Ping-pong select will come from DMMCPBPMPINPONSEL register.
18	DMMCPBPMPINPONSEL	R/W	0h	CP BPM Ping Pong select for HIL Mode
17	DMMADCBUFWREN	R/W	0h	ADC Buffer Write Enable from DMM. 0 --> Write to ADC BUF memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from ADCBUF_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMADCBUFPINPONSEL register.
16	DMMADCBUFPINPONSEL	R/W	0h	ADC Buffer Ping Pong select for HIL Mode
15-0	NU1	R	0h	Not Used

4.9.4.69 DSSINTRCFG Register (Offset = 270h) [reset = 0h]

DSSINTRCFG is shown in Figure 4-954 and described in Table 4-997.

Return to Summary Table.

Figure 4-954. DSSINTRCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
LGFRAMESTRTINTMUXSEL		PINPONINTMUXSEL		CHIRPAVLINTMUXSEL		FRAMESTRTINTMUXSEL	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 4-997. DSSINTRCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Not Used
7-6	LGFRAMESTRTINTMUXSEL	R/W	0h	[0] - Used to select between DFE and DMM Global Logical Frame Start CFG Bit. 0 --> Select DFE Logical Frame Start, 1--> Select DMM Global Logical Frame start CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 3.
5-4	PINPONINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Ping Pong Switch, 1--> Select DMM Global Ping Pong Switch CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 2.
3-2	CHIRPAVLINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Chirp Available, 1--> Select DMM Global Chirp Available CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 1.
1-0	FRAMESTRTINTMUXSEL	R/W	0h	[0] - Used to select between VIN/DFE and DMM Global Frame Start CFG Bit. 0 --> Select VIN/DFE Frame Start, 1--> Select DMM Global Frame start CFG Bit. [1] - Used to select between the result from above mentioned Mux and DMM SW Interrupt 0. 0 --> Selects the result of above mentioned mux. 1 --> Selects DMM SW Interrupt 0.

4.9.4.70 MPUMSTIDCFG1 Register (Offset = 274h) [reset = 1A191514h]

MPUMSTIDCFG1 is shown in Figure 4-955 and described in Table 4-998.

Return to Summary Table.

Figure 4-955. MPUMSTIDCFG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUMSTID3								MPUMSTID2							
R/W-1Ah								R/W-19h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPUMSTID1								MPUMSTID0							
R/W-15h								R/W-14h							

Table 4-998. MPUMSTIDCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MPUMSTID3	R/W	1Ah	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID3. Defaule value maps to RS232 Port
23-16	MPUMSTID2	R/W	19h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID2. Defaule value maps to MSS DAP Port
15-8	MPUMSTID1	R/W	15h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID0. Defaule value maps to MSS CR4 Read Port
7-0	MPUMSTID0	R/W	14h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID0. Defaule value maps to MSS CR4 Read Port

4.9.4.71 MPUMSTIDCFG2 Register (Offset = 278h) [reset = 1A191514h]

MPUMSTIDCFG2 is shown in Figure 4-956 and described in Table 4-999.

Return to Summary Table.

Figure 4-956. MPUMSTIDCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MPUMSTID7								MPUMSTID6							
R/W-1Ah								R/W-19h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPUMSTID5								MPUMSTID4							
R/W-15h								R/W-14h							

Table 4-999. MPUMSTIDCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MPUMSTID7	R/W	1Ah	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID7. Defaule value maps to RS232 Port
23-16	MPUMSTID6	R/W	19h	Allowed MSTID6. Defaule value maps to MSS DAP Port
15-8	MPUMSTID5	R/W	15h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID5 Defaule value maps to MSS CR4 Write Port
7-0	MPUMSTID4	R/W	14h	MPU on the MSS -> DSS CFG. Only Masters with Master ID configured in MPUMSTID[0-7] are allowed to access the DSS CFG Space. Allowed MSTID4. Defaule value maps to MSS CR4 Read Port

4.9.4.72 MPUMSTIDCFG3 Register (Offset = 27Ch) [reset = FFh]

MPUMSTIDCFG3 is shown in Figure 4-957 and described in Table 4-1000.

Return to Summary Table.

Figure 4-957. MPUMSTIDCFG3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				MPUMSTIDEN	RESERVED	MPUERRCLR	RESERVED
R/W-0h				R/W-0h	R/W-0h	0h	0h
15	14	13	12	11	10	9	8
MPUERRMSTID							
R-0h							
7	6	5	4	3	2	1	0
MPUMSTIDVLD							
R/W-FFh							

Table 4-1000. MPUMSTIDCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	MPUMSTIDEN	R/W	0h	Enable control for Master ID based MPU 0 --> Disabled, 1 --> Enabled
18	RESERVED	R/W	0h	Reserved
17	MPUERRCLR		0h	Error clear pulse for Master ID based MPU Write 0x1 to clear the previous error status: it is a wspecial access type; a write to this field generates a pulse.
16	RESERVED		0h	Reserved
15-8	MPUERRMSTID	R	0h	Error status field. Provides the Master ID which is not part of the allowed list which caused an error.
7-0	MPUMSTIDVLD	R/W	FFh	Master ID valid. Each bit corresponds to the MPUMSTID[7:0] 0 : Master ID entry is valid 1 : Master ID entry is not valid and entry does not have access to DSS CFG Space

4.9.4.73 HSRAM1ECCCFG Register (Offset = 280h) [reset = 0h]

HSRAM1ECCCFG is shown in Figure 4-958 and described in Table 4-1001.

Return to Summary Table.

Figure 4-958. HSRAM1ECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	HSRAM1ECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
HSRAM1ECCREPAIREDBIT	HSRAM1ECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
HSRAM1ECCFAULTADDRESS				HSRAM1ECCERRCLR	HSRAM1ECCEN	HSRAM1ECCINITDONE	HSRAM1ECCINIT
R-0h				0h	R/W-0h	R-0h	0h

Table 4-1001. HSRAM1ECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	HSRAM1ECCREPAIREDBIT	R	0h	Bit position of the repaired bit in HSRAM1
14-4	HSRAM1ECCFAULTADDRESS	R	0h	ECC Fault address in HSRAM1
3	HSRAM1ECCERRCLR		0h	Clear bit for ECC Error Indication in HSRAM1 Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	HSRAM1ECCEN	R/W	0h	Enable for ECC in HSRAM1
1	HSRAM1ECCINITDONE	R	0h	Done status for ECC Init for HSRAM1
0	HSRAM1ECCINIT		0h	ECC Init For HSRAM1: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.74 DATATRRAMECCCFG Register (Offset = 288h) [reset = 0h]

DATATRRAMECCCFG is shown in Figure 4-959 and described in Table 4-1002.

Return to Summary Table.

Figure 4-959. DATATRRAMECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU			DATATRRAMECCREPAIREDBIT				
R/W-0h			R-0h				
15	14	13	12	11	10	9	8
DATATRRAMECCREPAIREDBIT			DATATRRAMECCFAULTADDRESS				
R-0h			R-0h				
7	6	5	4	3	2	1	0
DATATRRAMECCFAULTADDRESS				DATATRRAMECCERRCLR	DATATRRAMECCEN	DATATRRAMECCINITDONE	DATATRRAMECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-1002. DATATRRAMECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU	R	0h	Not Used
20-13	DATATRRAMECCREPAIREDBIT	R	0h	Bit position of the repaired bit in DATATRRAM
12-4	DATATRRAMECCFAULTADDRESS	R	0h	ECC Fault address in DATATRRAM
3	DATATRRAMECCERRCLR	W	0h	Clear bit for ECC Error Indication in DATATRRAM Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	DATATRRAMECCEN	R/W	0h	Enable for ECC in DATATRRAM
1	DATATRRAMECCINITDONE	R	0h	Done status for ECC Init for Data Transfer RAM
0	DATATRRAMECCINIT	W	0h	ECC Init For Data Transfer RAM: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.75 ADCBUFPINGECCCFG Register (Offset = 28Ch) [reset = 0h]

ADCBUFPINGECCCFG is shown in Figure 4-960 and described in Table 4-1003.

Return to Summary Table.

Figure 4-960. ADCBUFPINGECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	ADCBUFPINGECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
ADCBUFPINGECCREPAIREDBIT	ADCBUFPINGECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
ADCBUFPINGECCFAULTADDRESS				ADCBUFPINGECCERRCLR	ADCBUFPINGECCEN	ADCBUFPINGECCINITDONE	ADCBUFPINGECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-1003. ADCBUFPINGECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	ADCBUFPINGECCREPAIREDBIT	R	0h	Bit position of the repaired bit in ADC Buffer Ping Memory
14-4	ADCBUFPINGECCFAULTADDRESS	R	0h	ECC Fault address in ADC Buffer Ping Memory
3	ADCBUFPINGECCERRCLR	W	0h	Clear bit for ECC Error Indication in ADC Buffer Ping Memory Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	ADCBUFPINGECCEN	R/W	0h	Enable for ECC in ADC Buffer Ping Memory
1	ADCBUFPINGECCINITDONE	R	0h	Done status for ECC Init for ADC Buffer Ping Memory
0	ADCBUFPINGECCINIT	W	0h	ECC Init For ADC Buffer Ping Memory: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.76 ADCBUFPONGECCCFG Register (Offset = 290h) [reset = 0h]

ADCBUFPONGECCCFG is shown in Figure 4-961 and described in Table 4-1004.

Return to Summary Table.

Figure 4-961. ADCBUFPONGECCCFG Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU	ADCBUFPONGECCREPAIREDBIT						
R/W-0h	R-0h						
15	14	13	12	11	10	9	8
ADCBUFPONGECCREPAIREDBIT	ADCBUFPONGECCFAULTADDRESS						
R-0h	R-0h						
7	6	5	4	3	2	1	0
ADCBUFPONGECCFAULTADDRESS				ADCBUFPONGECCERRCLR	ADCBUFPONGECCEN	ADCBUFPONGECCINITDONE	ADCBUFPONGECCINIT
R-0h				W-0h	R/W-0h	R-0h	W-0h

Table 4-1004. ADCBUFPONGECCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	Not Used
22-15	ADCBUFPONGECCREPAIREDBIT	R	0h	Bit position of the repaired bit in ADC Buffer Pong Memory
14-4	ADCBUFPONGECCFAULTADDRESS	R	0h	ECC Fault address in ADC Buffer Pong Memory
3	ADCBUFPONGECCERRCLR	W	0h	Clear bit for ECC Error Indication in ADC Buffer Pong Memory Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
2	ADCBUFPONGECCEN	R/W	0h	Enable for ECC in ADC Buffer Pong Memory
1	ADCBUFPONGECCINITDONE	R	0h	Done status for ECC Init for ADC Buffer Pong Memory
0	ADCBUFPONGECCINIT	W	0h	ECC Init For ADC Buffer Pong Memory: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.77 UMAP0PARITYCFG1 Register (Offset = 29Ch) [reset = 0h]

UMAP0PARITYCFG1 is shown in Figure 4-962 and described in Table 4-1005.

Return to Summary Table.

Figure 4-962. UMAP0PARITYCFG1 Register

31	30	29	28	27	26	25	24
NU						UMAP0BANK23ADDOUT	
R/W-0h						R-0h	
23	22	21	20	19	18	17	16
UMAP0BANK23ADDOUT							
R-0h							
15	14	13	12	11	10	9	8
UMAP0BANK23ADDOUT	UMAP0BANK01ADDOUT						
R-0h	R-0h						
7	6	5	4	3	2	1	0
UMAP0BANK01ADDOUT				UMAP0BANK23ERROUT	UMAP0BANK01ERROUT	UMAP0PARERRCLR	UMAP0PAREN
R-0h				R-0h	R-0h	0h	R/W-0h

Table 4-1005. UMAP0PARITYCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU	R/W	0h	Not Used
25-15	UMAP0BANK23ADDOUT	R	0h	Address corresponding to the parity error in Bank2 or Bank3 of UMAP0.
14-4	UMAP0BANK01ADDOUT	R	0h	Address corresponding to the parity error in Bank0 or Bank1 of UMAP0
3	UMAP0BANK23ERROUT	R	0h	Parity Error indication from either Bank2 or Bank3 of UMAP0
2	UMAP0BANK01ERROUT	R	0h	Parity Error indication from either Bank 0 or Bank1 of UMAP0
1	UMAP0PARERRCLR		0h	Clear pulse for all the error status from UMAP0 parity check logic. Self-clearing pulse Write 0x1 to clear the error status
0	UMAP0PAREN	R/W	0h	Enable for UMAP0 Parity Check logic. Assumed to be static. 0 --> Disable 1 --> Enable

4.9.4.78 UMAP0PARITYCFG2 Register (Offset = 2A0h) [reset = 0h]

UMAP0PARITYCFG2 is shown in Figure 4-963 and described in Table 4-1006.

Return to Summary Table.

Figure 4-963. UMAP0PARITYCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP0BANK1BITOUT																UMAP0BANK0BITOUT															
R-0h																R-0h															

Table 4-1006. UMAP0PARITYCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP0BANK1BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank1.
15-0	UMAP0BANK0BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank0.

4.9.4.79 UMAP0PARITYCFG3 Register (Offset = 2A4h) [reset = 0h]

UMAP0PARITYCFG3 is shown in Figure 4-964 and described in Table 4-1007.

Return to Summary Table.

Figure 4-964. UMAP0PARITYCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP0BANK3BITOUT																UMAP0BANK2BITOUT															
R-0h																R-0h															

Table 4-1007. UMAP0PARITYCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP0BANK3BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank3.
15-0	UMAP0BANK2BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP0 Bank2.

4.9.4.80 UMAP1PARITYCFG1 Register (Offset = 2A8h) [reset = 0h]

UMAP1PARITYCFG1 is shown in Figure 4-965 and described in Table 4-1008.

Return to Summary Table.

Figure 4-965. UMAP1PARITYCFG1 Register

31	30	29	28	27	26	25	24
NU						UMAP1BANK23ADDOUT	
R-0h						R-0h	
23	22	21	20	19	18	17	16
UMAP1BANK23ADDOUT							
R-0h							
15	14	13	12	11	10	9	8
UMAP1BANK23ADDOUT	UMAP1BANK01ADDOUT						
R-0h	R-0h						
7	6	5	4	3	2	1	0
UMAP1BANK01ADDOUT				UMAP1BANK23ERROUT	UMAP1BANK01ERROUT	UMAP1PARERRCLR	UMAP1PAREN
R-0h				R-0h	R-0h	W-0h	R/W-0h

Table 4-1008. UMAP1PARITYCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU	R	0h	Not Used
25-15	UMAP1BANK23ADDOUT	R	0h	Address corresponding to the parity error in Bank2 or Bank3 of UMAP1.
14-4	UMAP1BANK01ADDOUT	R	0h	Address corresponding to the parity error in Bank0 or Bank1 of UMAP1
3	UMAP1BANK23ERROUT	R	0h	Parity Error indication from either Bank2 or Bank3 of UMAP1
2	UMAP1BANK01ERROUT	R	0h	Parity Error indication from either Bank 0 or Bank1 of UMAP1
1	UMAP1PARERRCLR	W	0h	Clear pulse for all the error status from UMAP1 parity check logic. Self-clearing pulse Write 0x1 to clear the error status: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP1PAREN	R/W	0h	Enable for UMAP1 Parity Check logic. Assumed to be static. 0 --> Disable 1 --> Enable

4.9.4.81 UMAP1PARITYCFG2 Register (Offset = 2ACh) [reset = 0h]

UMAP1PARITYCFG2 is shown in Figure 4-966 and described in Table 4-1009.

Return to Summary Table.

Figure 4-966. UMAP1PARITYCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP1BANK1BITOUT																UMAP1BANK0BITOUT															
R-0h																R-0h															

Table 4-1009. UMAP1PARITYCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP1BANK1BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank1.
15-0	UMAP1BANK0BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank0.

4.9.4.82 UMAP1PARITYCFG3 Register (Offset = 2B0h) [reset = 0h]

UMAP1PARITYCFG3 is shown in Figure 4-967 and described in Table 4-1010.

Return to Summary Table.

Figure 4-967. UMAP1PARITYCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UMAP1BANK3BITOUT																UMAP1BANK2BITOUT															
R-0h																R-0h															

Table 4-1010. UMAP1PARITYCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UMAP1BANK3BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank3.
15-0	UMAP1BANK2BITOUT	R	0h	Bit level indication corresponding to parity error from UMAP1 Bank2.

4.9.4.83 ESMGRP2MASKCFG Register (Offset = 2B4h) [reset = FFFFFFFFh]

ESMGRP2MASKCFG is shown in Figure 4-968 and described in Table 4-1011.

Return to Summary Table.

Figure 4-968. ESMGRP2MASKCFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESMGRP2MASK																															
R/W-FFFFFFFFh																															

Table 4-1011. ESMGRP2MASKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESMGRP2MASK	R/W	FFFFFFFFh	Bbit level mask for each of the error signal connected to ESM Group2 input.

4.9.4.84 L2MEMINITCFG1 Register (Offset = 2B8h) [reset = 0h]

L2MEMINITCFG1 is shown in Figure 4-969 and described in Table 4-1012.

Return to Summary Table.

Figure 4-969. L2MEMINITCFG1 Register

31	30	29	28	27	26	25	24
UMAP1BANK3 PARINITDONE	UMAP1BANK2 PARINITDONE	UMAP1BANK1 PARINITDONE	UMAP1BANK0 PARINITDONE	UMAP0BANK3 PARINITDONE	UMAP0BANK2 PARINITDONE	UMAP0BANK1 PARINITDONE	UMAP0BANK0 PARINITDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
UMAP1BANK3 DATAINITDONE	UMAP1BANK2 DATAINITDONE	UMAP1BANK1 DATAINITDONE	UMAP1BANK0 DATAINITDONE	UMAP0BANK3 DATAINITDONE	UMAP0BANK2 DATAINITDONE	UMAP0BANK1 DATAINITDONE	UMAP0BANK0 DATAINITDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
UMAP1BANK3 PARINIT	UMAP1BANK2 PARINIT	UMAP1BANK1 PARINIT	UMAP1BANK0 PARINIT	UMAP0BANK3 PARINIT	UMAP0BANK2 PARINIT	UMAP0BANK1 PARINIT	UMAP0BANK0 PARINIT
0h	0h	0h	0h	0h	0h	0h	0h
7	6	5	4	3	2	1	0
UMAP1BANK3 DATAINIT	UMAP1BANK2 DATAINIT	UMAP1BANK1 DATAINIT	UMAP1BANK0 DATAINIT	UMAP0BANK3 DATAINIT	UMAP0BANK2 DATAINIT	UMAP0BANK1 DATAINIT	UMAP0BANK0 DATAINIT
0h	0h	0h	0h	0h	0h	0h	0h

Table 4-1012. L2MEMINITCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	UMAP1BANK3PARINITDONE	R	0h	Init Done status from UMAP1 Bank3 Parity memory
30	UMAP1BANK2PARINITDONE	R	0h	Init Done status from UMAP1 Bank2 Parity memory
29	UMAP1BANK1PARINITDONE	R	0h	Init Done status from UMAP1 Bank1 Parity memory
28	UMAP1BANK0PARINITDONE	R	0h	Init Done status from UMAP1 Bank0 Parity memory
27	UMAP0BANK3PARINITDONE	R	0h	Init Done status from UMAP0 Bank3 Parity memory
26	UMAP0BANK2PARINITDONE	R	0h	Init Done status from UMAP0 Bank2 Parity memory
25	UMAP0BANK1PARINITDONE	R	0h	Init Done status from UMAP0 Bank1 Parity memory
24	UMAP0BANK0PARINITDONE	R	0h	Init Done status from UMAP0 Bank0 Parity memory
23	UMAP1BANK3DATAINITDONE	R	0h	Init Done status from UMAP1 Bank3 Data memory
22	UMAP1BANK2DATAINITDONE	R	0h	Init Done status from UMAP1 Bank2 Data memory
21	UMAP1BANK1DATAINITDONE	R	0h	Init Done status from UMAP1 Bank1 Data memory
20	UMAP1BANK0DATAINITDONE	R	0h	Init Done status from UMAP1 Bank0 Data memory
19	UMAP0BANK3DATAINITDONE	R	0h	Init Done status from UMAP0 Bank3 Data memory
18	UMAP0BANK2DATAINITDONE	R	0h	Init Done status from UMAP0 Bank2 Data memory
17	UMAP0BANK1DATAINITDONE	R	0h	Init Done status from UMAP0 Bank1 Data memory

Table 4-1012. L2MEMINITCFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	UMAP0BANK0DATAINITDONE	R	0h	Init Done status from UMAP0 Bank0 Data memory
15	UMAP1BANK3PARINIT		0h	Init trigger for UMAP1 Bank3 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
14	UMAP1BANK2PARINIT		0h	Init trigger for UMAP1 Bank2 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
13	UMAP1BANK1PARINIT		0h	Init trigger for UMAP1 Bank1 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
12	UMAP1BANK0PARINIT		0h	Init trigger for UMAP1 Bank0 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
11	UMAP0BANK3PARINIT		0h	Init trigger for UMAP0 Bank3 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
10	UMAP0BANK2PARINIT		0h	Init trigger for UMAP0 Bank2 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
9	UMAP0BANK1PARINIT		0h	Init trigger for UMAP0 Bank1 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
8	UMAP0BANK0PARINIT		0h	Init trigger for UMAP0 Bank0 Parity memory: it is a wspecial access type; a write to this field generates a pulse.
7	UMAP1BANK3DATAINIT		0h	Init trigger for UMAP1 Bank3 Data memory: it is a wspecial access type; a write to this field generates a pulse.
6	UMAP1BANK2DATAINIT		0h	Init trigger for UMAP1 Bank2 Data memory: it is a wspecial access type; a write to this field generates a pulse.
5	UMAP1BANK1DATAINIT		0h	Init trigger for UMAP1 Bank1 Data memory: it is a wspecial access type; a write to this field generates a pulse.
4	UMAP1BANK0DATAINIT		0h	Init trigger for UMAP1 Bank0 Data memory: it is a wspecial access type; a write to this field generates a pulse.
3	UMAP0BANK3DATAINIT		0h	Init trigger for UMAP0 Bank3 Data memory: it is a wspecial access type; a write to this field generates a pulse.
2	UMAP0BANK2DATAINIT		0h	Init trigger for UMAP0 Bank2 Data memory: it is a wspecial access type; a write to this field generates a pulse.
1	UMAP0BANK1DATAINIT		0h	Init trigger for UMAP0 Bank1 Data memory: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP0BANK0DATAINIT		0h	Init trigger for UMAP0 Bank0 Data memory: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.85 L2MEMINITCFG2 Register (Offset = 2BCh) [reset = 0h]

L2MEMINITCFG2 is shown in Figure 4-970 and described in Table 4-1013.

Return to Summary Table.

Figure 4-970. L2MEMINITCFG2 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
UMAP1BANK1PRAMINITDONE	UMAP1BANK0PRAMINITDONE	UMAP0BANK1PRAMINITDONE	UMAP0BANK0PRAMINITDONE	UMAP1BANK1PRAMINIT	UMAP1BANK0PRAMINIT	UMAP0BANK1PRAMINIT	UMAP0BANK0PRAMINIT
R-0h	R-0h	R-0h	R-0h	W-0h	W-0h	W-0h	W-0h

Table 4-1013. L2MEMINITCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Not Used
7	UMAP1BANK1PRAMINITDONE	R	0h	Init Done Status for UMAP1 Bank1 PRAM memory
6	UMAP1BANK0PRAMINITDONE	R	0h	Init Done Status for UMAP1 Bank0 PRAM memory
5	UMAP0BANK1PRAMINITDONE	R	0h	Init Done Status for UMAP0 Bank1 PRAM memory
4	UMAP0BANK0PRAMINITDONE	R	0h	Init Done Status for UMAP0 Bank0 PRAM memory
3	UMAP1BANK1PRAMINIT	W	0h	Init trigger for UMAP1 Bank1 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
2	UMAP1BANK0PRAMINIT	W	0h	Init trigger for UMAP1 Bank0 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
1	UMAP0BANK1PRAMINIT	W	0h	Init trigger for UMAP0 Bank1 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.
0	UMAP0BANK0PRAMINIT	W	0h	Init trigger for UMAP0 Bank0 PRAM memory: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.86 GEMRSTCAUSE Register (Offset = 2C0h) [reset = 00010101h]

GEMRSTCAUSE is shown in Figure 4-971 and described in Table 4-1014.

Return to Summary Table.

Figure 4-971. GEMRSTCAUSE Register

31	30	29	28	27	26	25	24
NU2							GEMRSTCAUSECLR
R-0h							0h
23	22	21	20	19	18	17	16
GEMPORCAUSE							
R-1h							
15	14	13	12	11	10	9	8
GEMGRSTCAUSE							
R-1h							
7	6	5	4	3	2	1	0
GEMLRSTCAUSE							
R-1h							

Table 4-1014. GEMRSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU2	R	0h	
24	GEMRSTCAUSECLR		0h	Write 0x1 to clear the reset cause register for any previous resets: it is a wspecial access type; a write to this field generates a pulse.
23-16	GEMPORCAUSE	R	1h	DSP POR reset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMPORZ Bit 3 : Reset from Power FSM Bit 4 : Reset from STC FSM
15-8	GEMGRSTCAUSE	R	1h	DSP Greset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMGRSTN Bit 3 : Reset from Power FSM Bit 4 : Reset from STC FSM
7-0	GEMLRSTCAUSE	R	1h	DSP Lreset Bitwise Indication : Bit 0 : Por Reset Bit 1 : Warm Reset from TOPRCM Bit 2 : Reset from TOPRCM:DSSCTL.GEMLRSTN Bit 3 : Reset from Debugss Bit 4 : Reset from Power FSM Bit 5 : Reset from STC FSM

4.9.4.87 GEMPWRSMCFG4 Register (Offset = 2CCh) [reset = 00060000h]

GEMPWRSMCFG4 is shown in Figure 4-972 and described in Table 4-1015.

Return to Summary Table.

Figure 4-972. GEMPWRSMCFG4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					GEMEVENTMASK	PWRSMLRSTHALT	PWRSMSLEEPTRIG
R-0h					R/W-1h	R/W-1h	0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 4-1015. GEMPWRSMCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18	GEMEVENTMASK	R/W	1h	Mask bit for events going to DSP. When this bit is set (during GEM sleep/power down mode), the events are monitored outside and will be available for DSP to read and clear them once GEM wakes-up. The monitored events can be read from PWRSMEVNTMONSTATx registers.
17	PWRSMLRSTHALT	R/W	1h	Signal to halt DSP Power cycle state machine before de-asserting LRST of DSP. This is used during code download for the first time power up.
16	PWRSMSLEEPTRIG		0h	Sleep mode trigger for DSP power down state machine. This is honoured only when DSP is in GEM_ON state: it is a wspecial access type; a write to this field generates a pulse.
15-0	RESERVED	R	0h	Reserved

4.9.4.88 PWRSMWAKEMASK0 Register (Offset = 2D4h) [reset = FFFFFFFFh]

PWRSMWAKEMASK0 is shown in Figure 4-973 and described in Table 4-1016.

Return to Summary Table.

Figure 4-973. PWRSMWAKEMASK0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK0																															
R/W-FFFFFFFFh																															

Table 4-1016. PWRSMWAKEMASK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK0	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [31:0] 1 --> Masked, 0 --> Unmasked.

4.9.4.89 PWRSMWAKEMASK1 Register (Offset = 2D8h) [reset = FFFFFFFFh]

PWRSMWAKEMASK1 is shown in Figure 4-974 and described in Table 4-1017.

Return to Summary Table.

Figure 4-974. PWRSMWAKEMASK1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK1																															
R/W-FFFFFFFFh																															

Table 4-1017. PWRSMWAKEMASK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK1	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [63:32] 1 --> Masked, 0 --> Unmasked.

4.9.4.90 PWRSMWAKEMASK2 Register (Offset = 2DCh) [reset = FFFFFFFFh]

PWRSMWAKEMASK2 is shown in Figure 4-975 and described in Table 4-1018.

Return to Summary Table.

Figure 4-975. PWRSMWAKEMASK2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKEMASK2																															
R/W-FFFFFFFFh																															

Table 4-1018. PWRSMWAKEMASK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKEMASK2	R/W	FFFFFFFFh	Bit level mask for each of the wakeup source bits [95:64] 1 --> Masked, 0 --> Unmasked.

4.9.4.91 PWRSMMISEVTMASK0 Register (Offset = 2E0h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK0 is shown in Figure 4-976 and described in Table 4-1019.

Return to Summary Table.

Figure 4-976. PWRSMMISEVTMASK0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK0																															
R/W-FFFFFFFFh																															

Table 4-1019. PWRSMMISEVTMASK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK0	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[31:0] 1 --> Masked, 0 --> Unmasked.

4.9.4.92 PWRSMMISEVTMASK1 Register (Offset = 2E4h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK1 is shown in Figure 4-977 and described in Table 4-1020.

Return to Summary Table.

Figure 4-977. PWRSMMISEVTMASK1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK1																															
R/W-FFFFFFFFh																															

Table 4-1020. PWRSMMISEVTMASK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK1	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[63:32] 1 --> Masked, 0 --> Unmasked.

4.9.4.93 PWRSMMISEVTMASK2 Register (Offset = 2E8h) [reset = FFFFFFFFh]

PWRSMMISEVTMASK2 is shown in Figure 4-978 and described in Table 4-1021.

Return to Summary Table.

Figure 4-978. PWRSMMISEVTMASK2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMMISEVTMASK2																															
R/W-FFFFFFFFh																															

Table 4-1021. PWRSMMISEVTMASK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMMISEVTMASK2	R/W	FFFFFFFFh	Bit level mask for each of the missed events before getting pushed into GEM. Corresponds to Event lines[95:64] 1 --> Masked, 0 --> Unmasked.

4.9.4.94 PWRSMWAKESRCSTAT0 Register (Offset = 2ECh) [reset = 0h]

PWRSMWAKESRCSTAT0 is shown in Figure 4-979 and described in Table 4-1022.

Return to Summary Table.

Figure 4-979. PWRSMWAKESRCSTAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT0																															
R-0h																															

Table 4-1022. PWRSMWAKESRCSTAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT0	R	0h	Wakeup source status bits [31:0]

4.9.4.95 PWRSMWAKESRCSTAT1 Register (Offset = 2F0h) [reset = 0h]

PWRSMWAKESRCSTAT1 is shown in Figure 4-980 and described in Table 4-1023.

Return to Summary Table.

Figure 4-980. PWRSMWAKESRCSTAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT1																															
R-0h																															

Table 4-1023. PWRSMWAKESRCSTAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT1	R	0h	Wakeup source status bits [63:32]

4.9.4.96 PWRSMWAKESRCSTAT2 Register (Offset = 320h) [reset = 0h]

PWRSMWAKESRCSTAT2 is shown in Figure 4-981 and described in Table 4-1024.

Return to Summary Table.

Figure 4-981. PWRSMWAKESRCSTAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTAT2																															
R-0h																															

Table 4-1024. PWRSMWAKESRCSTAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTAT2	R	0h	Wakeup source status bits [95:64]

4.9.4.97 PWRSMEVNTMONSTAT0 Register (Offset = 324h) [reset = 0h]

PWRSMEVNTMONSTAT0 is shown in Figure 4-982 and described in Table 4-1025.

Return to Summary Table.

Figure 4-982. PWRSMEVNTMONSTAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT0																															
R-0h																															

Table 4-1025. PWRSMEVNTMONSTAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT0	R	0h	Missed events monitor status bits [31:0]. This is monitored when the events going to DSP are masked by GEMEVENTMASK register.

4.9.4.98 PWRSMEVNTMONSTAT1 Register (Offset = 328h) [reset = 0h]

PWRSMEVNTMONSTAT1 is shown in Figure 4-983 and described in Table 4-1026.

Return to Summary Table.

Figure 4-983. PWRSMEVNTMONSTAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT1																															
R-0h																															

Table 4-1026. PWRSMEVNTMONSTAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT1	R	0h	Missed events monitor status bits [63:32]

4.9.4.99 PWRSMEVNTMONSTAT2 Register (Offset = 32Ch) [reset = 0h]

PWRSMEVNTMONSTAT2 is shown in Figure 4-984 and described in Table 4-1027.

Return to Summary Table.

Figure 4-984. PWRSMEVNTMONSTAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMEVNTMONSTAT2																															
R-0h																															

Table 4-1027. PWRSMEVNTMONSTAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMEVNTMONSTAT2	R	0h	Missed events monitor status bits [95:64]

4.9.4.100 PWRSMWAKESRCSTATCLR0 Register (Offset = 330h) [reset = 0h]

PWRSMWAKESRCSTATCLR0 is shown in Figure 4-985 and described in Table 4-1028.

Return to Summary Table.

Figure 4-985. PWRSMWAKESRCSTATCLR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR0																															
0h																															

Table 4-1028. PWRSMWAKESRCSTATCLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR0		0h	Clear bit for wakeup source status bits [31:0]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.101 PWRSMWAKESRCSTATCLR1 Register (Offset = 334h) [reset = 0h]

PWRSMWAKESRCSTATCLR1 is shown in Figure 4-986 and described in Table 4-1029.

Return to Summary Table.

Figure 4-986. PWRSMWAKESRCSTATCLR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR1																															
0h																															

Table 4-1029. PWRSMWAKESRCSTATCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR1		0h	Clear bit for wakeup source status bits [63:32]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.102 PWRSMWAKESRCSTATCLR2 Register (Offset = 338h) [reset = 0h]

PWRSMWAKESRCSTATCLR2 is shown in Figure 4-987 and described in Table 4-1030.

Return to Summary Table.

Figure 4-987. PWRSMWAKESRCSTATCLR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRSMWAKESRCSTATCLR2																															
0h																															

Table 4-1030. PWRSMWAKESRCSTATCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PWRSMWAKESRCSTATCLR2		0h	Clear bit for wakeup source status bits [95:64]. Write 0x1 to clear the corresponding status bit: it is a wspecial access type; a write to this field generates a pulse.

4.9.4.103 ADCBUFCFG1 Register (Offset = 33Ch) [reset = 00010000h]

ADCBUFCFG1 is shown in Figure 4-988 and described in Table 4-1031.

Return to Summary Table.

Figure 4-988. ADCBUFCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
ADCBUFCONTSTOPPL	ADCBUFCONTSTRTPL	ADCBUFCONTMODEEN	ADCBUFWRITEMODE	RESERVED		RX3EN	RX2EN
W-0h	W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RX1EN	RX0EN	ADCBUFIQSWAP	RESERVED		ADCBUFREALONLYMODE	RESERVED	
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 4-1031. ADCBUFCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved
15	ADCBUFCONTSTOPPL	W	0h	Stop Pulse for Continuous mode. The data capture will stop once this register is set. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
14	ADCBUFCONTSTRTPL	W	0h	Start Pulse for Continuous mode. The data capture will start from Address 0 once this register is set. All the other configurations like Enable, Sample Count are expected to be programmed before this pulse. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
13	ADCBUFCONTMODEEN	R/W	0h	Continuous mode enable for ADC Buffer. This is set when a fixed number of samples have to be stored in Ping/Pong and not depend on Chirp time-lines (Eg: Analog Lab characterization to stream out continuous data from DFE). Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
12	ADCBUFWRITEMODE	R/W	0h	This needs to be programmed to 0x1 in 16xx 0 --> Interleaved, 1 --> Non-interleaved
11-10	RESERVED	R/W	0h	Reserved
9	RX3EN	R/W	0h	Enable for Rx3 write
8	RX2EN	R/W	0h	Enable for Rx2 write
7	RX1EN	R/W	0h	Enable for Rx1 write
6	RX0EN	R/W	0h	Enable for Rx0 write
5	ADCBUFIQSWAP	R/W	0h	0 --> I is stored in LSB and Q is stored in MSB 1 --> Q is stored in LSB and I is stored in MSB
4-3	RESERVED	R/W	0h	Reserved
2	ADCBUFREALONLYMODE	R/W	0h	0-->Complex Data mode, 1-->Real data mode
1-0	RESERVED	R/W	0h	Reserved

4.9.4.104 ADCBUFCFG2 Register (Offset = 340h) [reset = 02000000h]

ADCBUFCFG2 is shown in Figure 4-989 and described in Table 4-1032.

Return to Summary Table.

Figure 4-989. ADCBUFCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					ADCBUFADDRX1										
R-0h					R/W-200h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					ADCBUFADDRX0										
R-0h					R/W-0h										

Table 4-1032. ADCBUFCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	
26-16	ADCBUFADDRX1	R/W	200h	128 bit Address offset to be added to the internal address pointer for Rx1 writes in Non-interleaved mode.
15-11	NU1	R	0h	
10-0	ADCBUFADDRX0	R/W	0h	128 bit Address offset to be added to the internal address pointer for Rx0 writes in Non-interleaved mode.

4.9.4.105 ADCBUFCFG3 Register (Offset = 344h) [reset = 06000400h]

ADCBUFCFG3 is shown in Figure 4-990 and described in Table 4-1033.

Return to Summary Table.

Figure 4-990. ADCBUFCFG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					ADCBUFADDRX3										
R-0h					R/W-600h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					ADCBUFADDRX2										
R-0h					R/W-400h										

Table 4-1033. ADCBUFCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	
26-16	ADCBUFADDRX3	R/W	600h	128 bit Address offset to be added to the internal address pointer for Rx3 writes in Non-interleaved mode.
15-11	NU1	R	0h	
10-0	ADCBUFADDRX2	R/W	400h	128 bit Address offset to be added to the internal address pointer for Rx2 writes in Non-interleaved mode.

4.9.4.106 ADCBUFCFG4 Register (Offset = 348h) [reset = 400h]

ADCBUFCFG4 is shown in Figure 4-991 and described in Table 4-1034.

Return to Summary Table.

Figure 4-991. ADCBUFCFG4 Register

31	30	29	28	27	26	25	24
RESERVED						ADCBUFNUMCHRPPONG	
R-0h						R/W-0h	
23	22	21	20	19	18	17	16
ADCBUFNUMCHRPPONG			ADCBUFNUMCHRPPING				
R/W-0h			R/W-0h				
15	14	13	12	11	10	9	8
ADCBUFSAMPCNT							
R/W-400h							
7	6	5	4	3	2	1	0
ADCBUFSAMPCNT							
R/W-400h							

Table 4-1034. ADCBUFCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-21	ADCBUFNUMCHRPPONG	R/W	0h	Number of chirps to be stored in Pong buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Ping
20-16	ADCBUFNUMCHRPPING	R/W	0h	Number of chirps to be stored in Ping buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Pong
15-0	ADCBUFSAMPCNT	R/W	400h	No of samples to store in each Ping and Pong register in continuous mode of ADC Buffer. In real only mode this refers to the number of real samples and in complex mode, this refers to number of complex samples. This refers to the number of samples per channel. This counter increments once for every new sample from DFE (as long as 1 or more channels are enabled). The max allowed value varies depending on other configurations (No of channels enabled and real/complex data). Continous mode is expected to be only used for CZ and ADC Buffer Testpattern mode

4.9.4.107 STCPBISTSMCFG1 Register (Offset = 34Ch) [reset = 18h]

STCPBISTSMCFG1 is shown in Figure 4-992 and described in Table 4-1035.

Return to Summary Table.

Figure 4-992. STCPBISTSMCFG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED			PBISTTESTSTATCLR	PBISTTESTSTAT		STCPBISTSMSTATE	
R-0h			W-0h	R-0h		R-0h	
15	14	13	12	11	10	9	8
STCPBISTSMSTATE				RESERVED			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED			STCPBISTCKSTPACKMASK	STCPBISTLRSTDASRTHALT	STCPBISTSMTRIG	STCPBISTEN	
R/W-0h			R/W-1h	R/W-1h	W-0h	R/W-0h	

Table 4-1035. STCPBISTSMCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	Reserved
20	PBISTTESTSTATCLR	W	0h	Clear bit for PBIST Status: this is a wspecial access type - awrite to this field generates a pulse.
19-18	PBISTTESTSTAT	R	0h	PBIST status from GEM. [0] - Fail Indication [1] - Done indication
17-12	STCPBISTSMSTATE	R	0h	Current state of STC PBIST state machine
11-5	RESERVED	R/W	0h	Reserved
4	STCPBISTCKSTPACKMASK	R/W	1h	Mask bit for ignoring the clock stop ack from GEM. This will be used for ignoring clock stop ack during boot-up. 1 --> Ignore clock stop ack from GEM. 0 --> Wait for clock stop ack from GEM after giving clock stop request.
3	STCPBISTLRSTDASRTHALT	R/W	1h	Configuration to halt the state machine before the final de-assertion of LRST to enable program download. 1 --> Halt, 0 --> Proceed.
2	STCPBISTSMTRIG	W	0h	Trigger pulse for the STC PBIST state machine. This is a self-clearing pulse.
1-0	STCPBISTEN	R/W	0h	Enable for PBIST and STC. 00 - Reserved, 01 --> STC only 10 --> PBIST only 11 --> PBIST followed by STC.

4.9.4.108 STCPBISTSMCFG2 Register (Offset = 350h) [reset = 2410h]

STCPBISTSMCFG2 is shown in Figure 4-993 and described in Table 4-1036.

Return to Summary Table.

Figure 4-993. STCPBISTSMCFG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							BCK2BCKSTCEN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED		GEMPBISTROMCLKSEL		GEMTMODEVLCTASRTCNT			
R-0h		R/W-2h		R/W-10h			
7	6	5	4	3	2	1	0
GEMTMODEVLCTASRTCNT		GEMTMODEVLCTDASRTCNT					
R/W-10h		R/W-10h					

Table 4-1036. STCPBISTSMCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	BCK2BCKSTCEN	R/W	0h	Enables back to Back STC.Needs to be set to 1 for self test
15-14	RESERVED	R	0h	Reserved
13-12	GEMPBISTROMCLKSEL	R/W	2h	Pbist_rom_clk_div_sel --> It is used to select the PBIST rom clock frequency . 2b00 : Div 1 (600Mhz) 2b01 : Div 2 (300Mhz) 2b10 : Div 3 (200Mhz) 2b11 : Div 4 (150Mhz)
11-6	GEMTMODEVLCTASRTCNT	R/W	10h	No of clocks (in terms of 200 MHz DSPSS Bus clock) after asserting GEM TMODE VLCT signal before proceeding to the next state. Max allowed value is 31.
5-0	GEMTMODEVLCTDASRTCNT	R/W	10h	No of clocks (in terms of 200 MHz DSPSS Bus clock) after De-asserting GEM TMODE VLCT signal before proceeding to the next state. Max allowed value is 31.

4.9.4.109 RTI2EVENTCAPTURESEL Register (Offset = 358h) [reset = 0h]

RTI2EVENTCAPTURESEL is shown in Figure 4-994 and described in Table 4-1037.

Return to Summary Table.

Figure 4-994. RTI2EVENTCAPTURESEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2									RTI2EVT1						
R-0h									R/W-0h						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1									RTI2EVT0						
R-0h									R/W-0h						

Table 4-1037. RTI2EVENTCAPTURESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU2	R	0h	
22-16	RTI2EVT1	R/W	0h	Used to Select the event to be captured for RTI2 Event1.
15-7	NU1	R	0h	
6-0	RTI2EVT0	R/W	0h	Used to Select the event to be captured for RTI2 Event0.

4.9.4.110 DSSMISC5 Register (Offset = 35Ch) [reset = 0h]

DSSMISC5 is shown in Figure 4-995 and described in Table 4-1038.

Return to Summary Table.

Figure 4-995. DSSMISC5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPCC1PARMEMINITDONE	TPCC0PARMEMINITDONE	TPCC1PARMEMINIT	TPCC0PARMEMINIT	CPBPMPIPOSELVAL	CPBPMPIPOSELCNT	CQPIPOSELVAL	CQPIPOSELCNT
R-0h	R-0h	W-0h	W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-1038. DSSMISC5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPCC1PARMEMINITDONE	R	0h	Mem init done status for the TPCC1 parity memory
6	TPCC0PARMEMINITDONE	R	0h	Mem init done status for the TPCC0 parity memory
5	TPCC1PARMEMINIT	W	0h	Mem init for the TPCC1 parity memory: this is a wspecial access type, a write to this field generates a pulse.
4	TPCC0PARMEMINIT	W	0h	Mem init for the TPCC0 parity memory: this is a wspecial access type, a write to this field generates a pulse.
3	CPBPMPIPOSELVAL	R/W	0h	Ping pong select override value for CPBPM Memory. 1 --> Read access from CPBPM_MEM Slave of DSS Interconnect will be routed to ping memory and write access from CPBPM_W write will be routed to pong memory. 0 --> Read access from CPBPM_MEM Slave of DSS Interconnect will be routed to pong memory and write access from CPBPM_W write will be routed to ping memory.
2	CPBPMPIPOSELCNT	R/W	0h	Ping pong select override control for CPBPM Memory. 0 --> Ping-pong select comes from HW FSM (same as the ping-pong select for ADC Buffer)/DMMCPBPMPINPONSEL 1 --> Ping pong select for CPBPM memory is taken from SW register (CPBPMPIPOSELVAL)
1	CQPIPOSELVAL	R/W	0h	Ping pong select override value for CQ Memory. 1 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to ping memory and write access from CQ_W/DFE write will be routed to pong memory. 0 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to pong memory and write access from CQ_W/DFE write will be routed to ping memory.
0	CQPIPOSELCNT	R/W	0h	Ping pong select override control for CQ Memory. 0 --> Ping-pong select comes from HW FSM (same as the ping-pong select for ADC Buffer)/DMMCQPINPONSEL 1 --> Ping pong select for CQ memory is taken from SW register (CQPIPOSELVAL)

4.9.5 DSS_REG2 Registers

Table 4-1039 lists the memory-mapped registers for the DSS_REG2. All register offset addresses not listed in Table 4-1039 should be considered as reserved locations and the register contents should not be modified.

Table 4-1039. DSS_REG2 Registers

Offset	Acronym	Register Name	Section
100h	TPTC2WRMPUSTADD0	TPTC2WRMPUSTADD0	Section 4.9.5.1
104h	TPTC2WRMPUSTADD1	TPTC2WRMPUSTADD1	Section 4.9.5.2
108h	TPTC2WRMPUSTADD2	TPTC2WRMPUSTADD2	Section 4.9.5.3
10Ch	TPTC2WRMPUSTADD3	TPTC2WRMPUSTADD3	Section 4.9.5.4
110h	TPTC2WRMPUSTADD4	TPTC2WRMPUSTADD4	Section 4.9.5.5
114h	TPTC2WRMPUSTADD5	TPTC2WRMPUSTADD5	Section 4.9.5.6
120h	TPTC2WRMPUENDADD0	TPTC2WRMPUENDADD0	Section 4.9.5.7
124h	TPTC2WRMPUENDADD1	TPTC2WRMPUENDADD1	Section 4.9.5.8
128h	TPTC2WRMPUENDADD2	TPTC2WRMPUENDADD2	Section 4.9.5.9
12Ch	TPTC2WRMPUENDADD3	TPTC2WRMPUENDADD3	Section 4.9.5.10
130h	TPTC2WRMPUENDADD4	TPTC2WRMPUENDADD4	Section 4.9.5.11
134h	TPTC2WRMPUENDADD5	TPTC2WRMPUENDADD5	Section 4.9.5.12
140h	TPTC2WRMPUERRADD	TPTC2WRMPUERRADD	Section 4.9.5.13
148h	TPTC2RDMPUSTADD0	TPTC2RDMPUSTADD0	Section 4.9.5.14
14Ch	TPTC2RDMPUSTADD1	TPTC2RDMPUSTADD1	Section 4.9.5.15
150h	TPTC2RDMPUSTADD2	TPTC2RDMPUSTADD2	Section 4.9.5.16
154h	TPTC2RDMPUSTADD3	TPTC2RDMPUSTADD3	Section 4.9.5.17
158h	TPTC2RDMPUSTADD4	TPTC2RDMPUSTADD4	Section 4.9.5.18
15Ch	TPTC2RDMPUSTADD5	TPTC2RDMPUSTADD5	Section 4.9.5.19
168h	TPTC2RDMPUENDADD0	TPTC2RDMPUENDADD0	Section 4.9.5.20
16Ch	TPTC2RDMPUENDADD1	TPTC2RDMPUENDADD1	Section 4.9.5.21
170h	TPTC2RDMPUENDADD2	TPTC2RDMPUENDADD2	Section 4.9.5.22
174h	TPTC2RDMPUENDADD3	TPTC2RDMPUENDADD3	Section 4.9.5.23
178h	TPTC2RDMPUENDADD4	TPTC2RDMPUENDADD4	Section 4.9.5.24
17Ch	TPTC2RDMPUENDADD5	TPTC2RDMPUENDADD5	Section 4.9.5.25
188h	TPTC2RDMPUERRADD	TPTC2RDMPUERRADD	Section 4.9.5.26
18Ch	TPTC3WRMPUSTADD0	TPTC3WRMPUSTADD0	Section 4.9.5.27
190h	TPTC3WRMPUSTADD1	TPTC3WRMPUSTADD1	Section 4.9.5.28
194h	TPTC3WRMPUSTADD2	TPTC3WRMPUSTADD2	Section 4.9.5.29
198h	TPTC3WRMPUSTADD3	TPTC3WRMPUSTADD3	Section 4.9.5.30
19Ch	TPTC3WRMPUSTADD4	TPTC3WRMPUSTADD4	Section 4.9.5.31
1A0h	TPTC3WRMPUSTADD5	TPTC3WRMPUSTADD5	Section 4.9.5.32
1ACh	TPTC3WRMPUENDADD0	TPTC3WRMPUENDADD0	Section 4.9.5.33
1B0h	TPTC3WRMPUENDADD1	TPTC3WRMPUENDADD1	Section 4.9.5.34
1B4h	TPTC3WRMPUENDADD2	TPTC3WRMPUENDADD2	Section 4.9.5.35
1B8h	TPTC3WRMPUENDADD3	TPTC3WRMPUENDADD3	Section 4.9.5.36
1BCh	TPTC3WRMPUENDADD4	TPTC3WRMPUENDADD4	Section 4.9.5.37
1C0h	TPTC3WRMPUENDADD5	TPTC3WRMPUENDADD5	Section 4.9.5.38
1CCh	TPTC3WRMPUERRADD	TPTC3WRMPUERRADD	Section 4.9.5.39
1D0h	TPTC3RDMPUSTADD0	TPTC3RDMPUSTADD0	Section 4.9.5.40
1D4h	TPTC3RDMPUSTADD1	TPTC3RDMPUSTADD1	Section 4.9.5.41
1D8h	TPTC3RDMPUSTADD2	TPTC3RDMPUSTADD2	Section 4.9.5.42
1DCh	TPTC3RDMPUSTADD3	TPTC3RDMPUSTADD3	Section 4.9.5.43

Table 4-1039. DSS_REG2 Registers (continued)

Offset	Acronym	Register Name	Section
1E0h	TPTC3RDMPUSTADD4	TPTC3RDMPUSTADD4	Section 4.9.5.44
1E4h	TPTC3RDMPUSTADD5	TPTC3RDMPUSTADD5	Section 4.9.5.45
1F0h	TPTC3RDMPUENDADD0	TPTC3RDMPUENDADD0	Section 4.9.5.46
1F4h	TPTC3RDMPUENDADD1	TPTC3RDMPUENDADD1	Section 4.9.5.47
1F8h	TPTC3RDMPUENDADD2	TPTC3RDMPUENDADD2	Section 4.9.5.48
1FCh	TPTC3RDMPUENDADD3	TPTC3RDMPUENDADD3	Section 4.9.5.49
200h	TPTC3RDMPUENDADD4	TPTC3RDMPUENDADD4	Section 4.9.5.50
204h	TPTC3RDMPUENDADD5	TPTC3RDMPUENDADD5	Section 4.9.5.51
210h	TPTC3RDMPUERRADD	TPTC3RDMPUERRADD	Section 4.9.5.52
214h	TPTCMPUVALIDCFG2	TPTCMPUVALIDCFG2	Section 4.9.5.53
218h	TPTCMPUENCFG2	TPTCMPUENCFG2	Section 4.9.5.54
268h	L3ECCCFG1	L3ECCCFG1	Section 4.9.5.55
26Ch	L3ECCCFG2	L3ECCCFG2	Section 4.9.5.56
270h	DSS2MSSSWIRQ		Section 4.9.5.57

Complex bit access types are encoded to fit into small table cells. Table 4-1040 shows the codes that are used for access types in this section.

Table 4-1040. DSS_REG2 Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

4.9.5.1 TPTC2WRMPUSTADD0 Register (Offset = 100h) [reset = 0h]

TPTC2WRMPUSTADD0 is shown in Figure 4-996 and described in Table 4-1041.

Return to Summary Table.

Figure 4-996. TPTC2WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD0																															
R/W-0h																															

Table 4-1041. TPTC2WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC2

4.9.5.2 TPTC2WRMPUSTADD1 Register (Offset = 104h) [reset = 0h]

TPTC2WRMPUSTADD1 is shown in Figure 4-997 and described in Table 4-1042.

Return to Summary Table.

Figure 4-997. TPTC2WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD1																															
R/W-0h																															

Table 4-1042. TPTC2WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC2

4.9.5.3 TPTC2WRMPUSTADD2 Register (Offset = 108h) [reset = 0h]

TPTC2WRMPUSTADD2 is shown in Figure 4-998 and described in Table 4-1043.

Return to Summary Table.

Figure 4-998. TPTC2WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD2																															
R/W-0h																															

Table 4-1043. TPTC2WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC2

4.9.5.4 TPTC2WRMPUSTADD3 Register (Offset = 10Ch) [reset = 0h]

TPTC2WRMPUSTADD3 is shown in Figure 4-999 and described in Table 4-1044.

Return to Summary Table.

Figure 4-999. TPTC2WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD3																															
R/W-0h																															

Table 4-1044. TPTC2WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC2

4.9.5.5 TPTC2WRMPUSTADD4 Register (Offset = 110h) [reset = 0h]

TPTC2WRMPUSTADD4 is shown in Figure 4-1000 and described in Table 4-1045.

Return to Summary Table.

Figure 4-1000. TPTC2WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD4																															
R/W-0h																															

Table 4-1045. TPTC2WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC2

4.9.5.6 TPTC2WRMPUSTADD5 Register (Offset = 114h) [reset = 0h]

TPTC2WRMPUSTADD5 is shown in Figure 4-1001 and described in Table 4-1046.

Return to Summary Table.

Figure 4-1001. TPTC2WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUSTADD5																															
R/W-0h																															

Table 4-1046. TPTC2WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC2

4.9.5.7 TPTC2WRMPUENDADD0 Register (Offset = 120h) [reset = 0h]

TPTC2WRMPUENDADD0 is shown in Figure 4-1002 and described in Table 4-1047.

Return to Summary Table.

Figure 4-1002. TPTC2WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD0																															
R/W-0h																															

Table 4-1047. TPTC2WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC2

4.9.5.8 TPTC2WRMPUENDADD1 Register (Offset = 124h) [reset = 0h]

TPTC2WRMPUENDADD1 is shown in Figure 4-1003 and described in Table 4-1048.

Return to Summary Table.

Figure 4-1003. TPTC2WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD1																															
R/W-0h																															

Table 4-1048. TPTC2WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC2

4.9.5.9 TPTC2WRMPUENDADD2 Register (Offset = 128h) [reset = 0h]

TPTC2WRMPUENDADD2 is shown in Figure 4-1004 and described in Table 4-1049.

Return to Summary Table.

Figure 4-1004. TPTC2WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD2																															
R/W-0h																															

Table 4-1049. TPTC2WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC2

4.9.5.10 TPTC2WRMPUENDADD3 Register (Offset = 12Ch) [reset = 0h]

TPTC2WRMPUENDADD3 is shown in Figure 4-1005 and described in Table 4-1050.

Return to Summary Table.

Figure 4-1005. TPTC2WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD3																															
R/W-0h																															

Table 4-1050. TPTC2WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC2

4.9.5.11 TPTC2WRMPUENDADD4 Register (Offset = 130h) [reset = 0h]

TPTC2WRMPUENDADD4 is shown in Figure 4-1006 and described in Table 4-1051.

Return to Summary Table.

Figure 4-1006. TPTC2WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD4																															
R/W-0h																															

Table 4-1051. TPTC2WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC2

4.9.5.12 TPTC2WRMPUENDADD5 Register (Offset = 134h) [reset = 0h]

TPTC2WRMPUENDADD5 is shown in Figure 4-1007 and described in Table 4-1052.

Return to Summary Table.

Figure 4-1007. TPTC2WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUENDADD5																															
R/W-0h																															

Table 4-1052. TPTC2WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC2

4.9.5.13 TPTC2WRMPUERRADD Register (Offset = 140h) [reset = 0h]

TPTC2WRMPUERRADD is shown in Figure 4-1008 and described in Table 4-1053.

Return to Summary Table.

Figure 4-1008. TPTC2WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2WRMPUERRADD																															
R-0h																															

Table 4-1053. TPTC2WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC2

4.9.5.14 TPTC2RDMPUSTADD0 Register (Offset = 148h) [reset = 0h]

TPTC2RDMPUSTADD0 is shown in Figure 4-1009 and described in Table 4-1054.

Return to Summary Table.

Figure 4-1009. TPTC2RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD0																															
R/W-0h																															

Table 4-1054. TPTC2RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC2

4.9.5.15 TPTC2RDMPUSTADD1 Register (Offset = 14Ch) [reset = 0h]

TPTC2RDMPUSTADD1 is shown in Figure 4-1010 and described in Table 4-1055.

Return to Summary Table.

Figure 4-1010. TPTC2RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD1																															
R/W-0h																															

Table 4-1055. TPTC2RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC2

4.9.5.16 TPTC2RDMPUSTADD2 Register (Offset = 150h) [reset = 0h]

TPTC2RDMPUSTADD2 is shown in Figure 4-1011 and described in Table 4-1056.

Return to Summary Table.

Figure 4-1011. TPTC2RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD2																															
R/W-0h																															

Table 4-1056. TPTC2RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC2

4.9.5.17 TPTC2RDMPUSTADD3 Register (Offset = 154h) [reset = 0h]

TPTC2RDMPUSTADD3 is shown in Figure 4-1012 and described in Table 4-1057.

Return to Summary Table.

Figure 4-1012. TPTC2RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD3																															
R/W-0h																															

Table 4-1057. TPTC2RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC2

4.9.5.18 TPTC2RDMPUSTADD4 Register (Offset = 158h) [reset = 0h]

TPTC2RDMPUSTADD4 is shown in Figure 4-1013 and described in Table 4-1058.

Return to Summary Table.

Figure 4-1013. TPTC2RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD4																															
R/W-0h																															

Table 4-1058. TPTC2RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC2

4.9.5.19 TPTC2RDMPUSTADD5 Register (Offset = 15Ch) [reset = 0h]

TPTC2RDMPUSTADD5 is shown in Figure 4-1014 and described in Table 4-1059.

Return to Summary Table.

Figure 4-1014. TPTC2RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUSTADD5																															
R/W-0h																															

Table 4-1059. TPTC2RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC2

4.9.5.20 TPTC2RDMPUENDADD0 Register (Offset = 168h) [reset = 0h]

TPTC2RDMPUENDADD0 is shown in Figure 4-1015 and described in Table 4-1060.

Return to Summary Table.

Figure 4-1015. TPTC2RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD0																															
R/W-0h																															

Table 4-1060. TPTC2RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC2

4.9.5.21 TPTC2RDMPUENDADD1 Register (Offset = 16Ch) [reset = 0h]

TPTC2RDMPUENDADD1 is shown in Figure 4-1016 and described in Table 4-1061.

Return to Summary Table.

Figure 4-1016. TPTC2RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD1																															
R/W-0h																															

Table 4-1061. TPTC2RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC2

4.9.5.22 TPTC2RDMPUENDADD2 Register (Offset = 170h) [reset = 0h]

TPTC2RDMPUENDADD2 is shown in Figure 4-1017 and described in Table 4-1062.

Return to Summary Table.

Figure 4-1017. TPTC2RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD2																															
R/W-0h																															

Table 4-1062. TPTC2RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC2

4.9.5.23 TPTC2RDMPUENDADD3 Register (Offset = 174h) [reset = 0h]

TPTC2RDMPUENDADD3 is shown in Figure 4-1018 and described in Table 4-1063.

Return to Summary Table.

Figure 4-1018. TPTC2RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD3																															
R/W-0h																															

Table 4-1063. TPTC2RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC2

4.9.5.24 TPTC2RDMPUENDADD4 Register (Offset = 178h) [reset = 0h]

TPTC2RDMPUENDADD4 is shown in Figure 4-1019 and described in Table 4-1064.

Return to Summary Table.

Figure 4-1019. TPTC2RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD4																															
R/W-0h																															

Table 4-1064. TPTC2RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC2

4.9.5.25 TPTC2RDMPUENDADD5 Register (Offset = 17Ch) [reset = 0h]

TPTC2RDMPUENDADD5 is shown in Figure 4-1020 and described in Table 4-1065.

Return to Summary Table.

Figure 4-1020. TPTC2RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUENDADD5																															
R/W-0h																															

Table 4-1065. TPTC2RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC2

4.9.5.26 TPTC2RDMPUERRADD Register (Offset = 188h) [reset = 0h]

TPTC2RDMPUERRADD is shown in Figure 4-1021 and described in Table 4-1066.

Return to Summary Table.

Figure 4-1021. TPTC2RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPUERRADD																															
R-0h																															

Table 4-1066. TPTC2RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC2RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC2

4.9.5.27 TPTC3WRMPUSTADD0 Register (Offset = 18Ch) [reset = 0h]

TPTC3WRMPUSTADD0 is shown in Figure 4-1022 and described in Table 4-1067.

Return to Summary Table.

Figure 4-1022. TPTC3WRMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD0																															
R/W-0h																															

Table 4-1067. TPTC3WRMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the write port of TPTC3

4.9.5.28 TPTC3WRMPUSTADD1 Register (Offset = 190h) [reset = 0h]

TPTC3WRMPUSTADD1 is shown in Figure 4-1023 and described in Table 4-1068.

Return to Summary Table.

Figure 4-1023. TPTC3WRMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD1																															
R/W-0h																															

Table 4-1068. TPTC3WRMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the write port of TPTC3

4.9.5.29 TPTC3WRMPUSTADD2 Register (Offset = 194h) [reset = 0h]

TPTC3WRMPUSTADD2 is shown in Figure 4-1024 and described in Table 4-1069.

Return to Summary Table.

Figure 4-1024. TPTC3WRMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD2																															
R/W-0h																															

Table 4-1069. TPTC3WRMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the write port of TPTC3

4.9.5.30 TPTC3WRMPUSTADD3 Register (Offset = 198h) [reset = 0h]

TPTC3WRMPUSTADD3 is shown in Figure 4-1025 and described in Table 4-1070.

Return to Summary Table.

Figure 4-1025. TPTC3WRMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD3																															
R/W-0h																															

Table 4-1070. TPTC3WRMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the write port of TPTC3

4.9.5.31 TPTC3WRMPUSTADD4 Register (Offset = 19Ch) [reset = 0h]

TPTC3WRMPUSTADD4 is shown in Figure 4-1026 and described in Table 4-1071.

Return to Summary Table.

Figure 4-1026. TPTC3WRMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD4																															
R/W-0h																															

Table 4-1071. TPTC3WRMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the write port of TPTC3

4.9.5.32 TPTC3WRMPUSTADD5 Register (Offset = 1A0h) [reset = 0h]

TPTC3WRMPUSTADD5 is shown in Figure 4-1027 and described in Table 4-1072.

Return to Summary Table.

Figure 4-1027. TPTC3WRMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUSTADD5																															
R/W-0h																															

Table 4-1072. TPTC3WRMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the write port of TPTC3

4.9.5.33 TPTC3WRMPUENDADD0 Register (Offset = 1ACh) [reset = 0h]

TPTC3WRMPUENDADD0 is shown in Figure 4-1028 and described in Table 4-1073.

Return to Summary Table.

Figure 4-1028. TPTC3WRMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD0																															
R/W-0h																															

Table 4-1073. TPTC3WRMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the write port of TPTC3

4.9.5.34 TPTC3WRMPUENDADD1 Register (Offset = 1B0h) [reset = 0h]

TPTC3WRMPUENDADD1 is shown in Figure 4-1029 and described in Table 4-1074.

Return to Summary Table.

Figure 4-1029. TPTC3WRMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD1																															
R/W-0h																															

Table 4-1074. TPTC3WRMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the write port of TPTC3

4.9.5.35 TPTC3WRMPUENDADD2 Register (Offset = 1B4h) [reset = 0h]

TPTC3WRMPUENDADD2 is shown in Figure 4-1030 and described in Table 4-1075.

Return to Summary Table.

Figure 4-1030. TPTC3WRMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD2																															
R/W-0h																															

Table 4-1075. TPTC3WRMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the write port of TPTC3

4.9.5.36 TPTC3WRMPUENDADD3 Register (Offset = 1B8h) [reset = 0h]

TPTC3WRMPUENDADD3 is shown in Figure 4-1031 and described in Table 4-1076.

Return to Summary Table.

Figure 4-1031. TPTC3WRMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD3																															
R/W-0h																															

Table 4-1076. TPTC3WRMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the write port of TPTC3

4.9.5.37 TPTC3WRMPUENDADD4 Register (Offset = 1BCh) [reset = 0h]

TPTC3WRMPUENDADD4 is shown in Figure 4-1032 and described in Table 4-1077.

Return to Summary Table.

Figure 4-1032. TPTC3WRMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD4																															
R/W-0h																															

Table 4-1077. TPTC3WRMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the write port of TPTC3

4.9.5.38 TPTC3WRMPUENDADD5 Register (Offset = 1C0h) [reset = 0h]

TPTC3WRMPUENDADD5 is shown in Figure 4-1033 and described in Table 4-1078.

Return to Summary Table.

Figure 4-1033. TPTC3WRMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUENDADD5																															
R/W-0h																															

Table 4-1078. TPTC3WRMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the write port of TPTC3

4.9.5.39 TPTC3WRMPUERRADD Register (Offset = 1CCh) [reset = 0h]

TPTC3WRMPUERRADD is shown in Figure 4-1034 and described in Table 4-1079.

Return to Summary Table.

Figure 4-1034. TPTC3WRMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3WRMPUERRADD																															
R-0h																															

Table 4-1079. TPTC3WRMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3WRMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the write port of TPTC3

4.9.5.40 TPTC3RDMPUSTADD0 Register (Offset = 1D0h) [reset = 0h]

TPTC3RDMPUSTADD0 is shown in Figure 4-1035 and described in Table 4-1080.

Return to Summary Table.

Figure 4-1035. TPTC3RDMPUSTADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD0																															
R/W-0h																															

Table 4-1080. TPTC3RDMPUSTADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD0	R/W	0h	Configure the Start address for Region 0 for the MPU on the read port of TPTC3

4.9.5.41 TPTC3RDMPUSTADD1 Register (Offset = 1D4h) [reset = 0h]

TPTC3RDMPUSTADD1 is shown in Figure 4-1036 and described in Table 4-1081.

Return to Summary Table.

Figure 4-1036. TPTC3RDMPUSTADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD1																															
R/W-0h																															

Table 4-1081. TPTC3RDMPUSTADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD1	R/W	0h	Configure the Start address for Region 1 for the MPU on the read port of TPTC3

4.9.5.42 TPTC3RDMPUSTADD2 Register (Offset = 1D8h) [reset = 0h]

TPTC3RDMPUSTADD2 is shown in Figure 4-1037 and described in Table 4-1082.

Return to Summary Table.

Figure 4-1037. TPTC3RDMPUSTADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD2																															
R/W-0h																															

Table 4-1082. TPTC3RDMPUSTADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD2	R/W	0h	Configure the Start address for Region 2 for the MPU on the read port of TPTC3

4.9.5.43 TPTC3RDMPUSTADD3 Register (Offset = 1DCh) [reset = 0h]

TPTC3RDMPUSTADD3 is shown in Figure 4-1038 and described in Table 4-1083.

Return to Summary Table.

Figure 4-1038. TPTC3RDMPUSTADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD3																															
R/W-0h																															

Table 4-1083. TPTC3RDMPUSTADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD3	R/W	0h	Configure the Start address for Region 3 for the MPU on the read port of TPTC3

4.9.5.44 TPTC3RDMPUSTADD4 Register (Offset = 1E0h) [reset = 0h]

TPTC3RDMPUSTADD4 is shown in Figure 4-1039 and described in Table 4-1084.

Return to Summary Table.

Figure 4-1039. TPTC3RDMPUSTADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD4																															
R/W-0h																															

Table 4-1084. TPTC3RDMPUSTADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD4	R/W	0h	Configure the Start address for Region 4 for the MPU on the read port of TPTC3

4.9.5.45 TPTC3RDMPUSTADD5 Register (Offset = 1E4h) [reset = 0h]

TPTC3RDMPUSTADD5 is shown in Figure 4-1040 and described in Table 4-1085.

Return to Summary Table.

Figure 4-1040. TPTC3RDMPUSTADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUSTADD5																															
R/W-0h																															

Table 4-1085. TPTC3RDMPUSTADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUSTADD5	R/W	0h	Configure the Start address for Region 5 for the MPU on the read port of TPTC3

4.9.5.46 TPTC3RDMPUENDADD0 Register (Offset = 1F0h) [reset = 0h]

TPTC3RDMPUENDADD0 is shown in Figure 4-1041 and described in Table 4-1086.

Return to Summary Table.

Figure 4-1041. TPTC3RDMPUENDADD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD0																															
R/W-0h																															

Table 4-1086. TPTC3RDMPUENDADD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD0	R/W	0h	Configure the End address for Region 0 for the MPU on the read port of TPTC3

4.9.5.47 TPTC3RDMPUENDADD1 Register (Offset = 1F4h) [reset = 0h]

TPTC3RDMPUENDADD1 is shown in Figure 4-1042 and described in Table 4-1087.

Return to Summary Table.

Figure 4-1042. TPTC3RDMPUENDADD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD1																															
R/W-0h																															

Table 4-1087. TPTC3RDMPUENDADD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD1	R/W	0h	Configure the End address for Region 1 for the MPU on the read port of TPTC3

4.9.5.48 TPTC3RDMPUENDADD2 Register (Offset = 1F8h) [reset = 0h]

TPTC3RDMPUENDADD2 is shown in Figure 4-1043 and described in Table 4-1088.

Return to Summary Table.

Figure 4-1043. TPTC3RDMPUENDADD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD2																															
R/W-0h																															

Table 4-1088. TPTC3RDMPUENDADD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD2	R/W	0h	Configure the End address for Region 2 for the MPU on the read port of TPTC3

4.9.5.49 TPTC3RDMPUENDADD3 Register (Offset = 1FCh) [reset = 0h]

TPTC3RDMPUENDADD3 is shown in Figure 4-1044 and described in Table 4-1089.

Return to Summary Table.

Figure 4-1044. TPTC3RDMPUENDADD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD3																															
R/W-0h																															

Table 4-1089. TPTC3RDMPUENDADD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD3	R/W	0h	Configure the End address for Region 3 for the MPU on the read port of TPTC3

4.9.5.50 TPTC3RDMPUENDADD4 Register (Offset = 200h) [reset = 0h]

TPTC3RDMPUENDADD4 is shown in Figure 4-1045 and described in Table 4-1090.

Return to Summary Table.

Figure 4-1045. TPTC3RDMPUENDADD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD4																															
R/W-0h																															

Table 4-1090. TPTC3RDMPUENDADD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD4	R/W	0h	Configure the End address for Region 4 for the MPU on the read port of TPTC3

4.9.5.51 TPTC3RDMPUENDADD5 Register (Offset = 204h) [reset = 0h]

TPTC3RDMPUENDADD5 is shown in Figure 4-1046 and described in Table 4-1091.

Return to Summary Table.

Figure 4-1046. TPTC3RDMPUENDADD5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUENDADD5																															
R/W-0h																															

Table 4-1091. TPTC3RDMPUENDADD5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUENDADD5	R/W	0h	Configure the End address for Region 5 for the MPU on the read port of TPTC3

4.9.5.52 TPTC3RDMPUERRADD Register (Offset = 210h) [reset = 0h]

TPTC3RDMPUERRADD is shown in Figure 4-1047 and described in Table 4-1092.

Return to Summary Table.

Figure 4-1047. TPTC3RDMPUERRADD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC3RDMPUERRADD																															
R-0h																															

Table 4-1092. TPTC3RDMPUERRADD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TPTC3RDMPUERRADD	R	0h	Status register to Read the address that triggered an MPU error on the read port of TPTC3

4.9.5.53 TPTCMPUVALIDCFG2 Register (Offset = 214h) [reset = 0h]

TPTCMPUVALIDCFG2 is shown in Figure 4-1048 and described in Table 4-1093.

Return to Summary Table.

Figure 4-1048. TPTCMPUVALIDCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPTC3RDMPURNGVLD								TPTC3WRMPURNGVLD							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPTC2RDMPURNGVLD								TPTC2WRMPURNGVLD							
R/W-0h								R/W-0h							

Table 4-1093. TPTCMPUVALIDCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TPTC3RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC3. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
23-16	TPTC3WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC3. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
15-8	TPTC2RDMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the read port of TPTC2. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled
7-0	TPTC2WRMPURNGVLD	R/W	0h	Configure the Valid bit for each address range for the MPU in the write port of TPTC2. [0]->Address0 and [5]-->Address5 Each bit corresponds to a MPU region 0 : Region is disbaled 1 : Regsion is enabled

4.9.5.54 TPTCMPUENCFG2 Register (Offset = 218h) [reset = 0h]

TPTCMPUENCFG2 is shown in Figure 4-1049 and described in Table 4-1094.

Return to Summary Table.

Figure 4-1049. TPTCMPUENCFG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPTC3RDMPUERRCLR	TPTC3WRMPUERRCLR	TPTC2RDMPUERRCLR	TPTC2WRMPUERRCLR	TPTC3RDMPUEN	TPTC3WRMPUEN	TPTC2RDMPUEN	TPTC2WRMPUEN
0h	0h	0h	0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 4-1094. TPTCMPUENCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TPTC3RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC3. Write 0x1 to clear the MPU error
6	TPTC3WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC3. Write 0x1 to clear the MPU error
5	TPTC2RDMPUERRCLR		0h	Clear flag for Error from the MPU in the read port of TPTC2. Write 0x1 to clear the MPU error
4	TPTC2WRMPUERRCLR		0h	Clear flag for Error from the MPU in the write port of TPTC2. Write 0x1 to clear the MPU error
3	TPTC3RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC3. 0 : MPU is disabled 1 : MPU is enabled
2	TPTC3WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC3. 0 : MPU is disabled 1 : MPU is enabled
1	TPTC2RDMPUEN	R/W	0h	Enable bit for the MPU in the read port of TPTC2. 0 : MPU is disabled 1 : MPU is enabled
0	TPTC2WRMPUEN	R/W	0h	Enable bit for the MPU in the write port of TPTC2. 0 : MPU is disabled 1 : MPU is enabled

4.9.5.55 L3ECCCFG1 Register (Offset = 268h) [reset = 0h]

L3ECCCFG1 is shown in Figure 4-1050 and described in Table 4-1095.

Return to Summary Table.

Figure 4-1050. L3ECCCFG1 Register

31	30	29	28	27	26	25	24
NU					L3ECCREPAIREDBIT		
R-0h					R-0h		
23	22	21	20	19	18	17	16
L3ECCREPAIREDBIT							
R-0h							
15	14	13	12	11	10	9	8
L3ECCREPAIREDBIT							
R-0h							
7	6	5	4	3	2	1	0
L3ECCREPAIREDBIT					L3ECCERRSTAT	L3ECCERRCLR	L3ECCEN
R-0h					R-0h	0h	R/W-0h

Table 4-1095. L3ECCCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU	R	0h	Not used
26-3	L3ECCREPAIREDBIT	R	0h	Bit position of repaired bit in L3 ECC memory. Each 6 bits out is this register maps to the corresponding 32 bit location in the data.
2	L3ECCERRSTAT	R	0h	Latched status for L3 ECC error.
1	L3ECCERRCLR		0h	Clear bit for L3 ECC.
0	L3ECCEN	R/W	0h	Enable for L3 ECC logic

4.9.5.56 L3ECCCFG2 Register (Offset = 26Ch) [reset = 0h]

L3ECCCFG2 is shown in Figure 4-1051 and described in Table 4-1096.

Return to Summary Table.

Figure 4-1051. L3ECCCFG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU															L3ECCFAULTADDR																
R-0h															R-0h																

Table 4-1096. L3ECCCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU	R	0h	Not used
16-0	L3ECCFAULTADDR	R	0h	Fault address of L3 ECC memory.

4.9.5.57 DSS2MSSSWIRQ Register (Offset = 270h) [reset = 0h]

DSS2MSSSWIRQ is shown in Figure 4-1052 and described in Table 4-1097.

Return to Summary Table.

Figure 4-1052. DSS2MSSSWIRQ Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU						MSSSWIRQ2	MSSSWIRQ1
R-0h						0h	0h

Table 4-1097. DSS2MSSSWIRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	0h	Not used
1	MSSSWIRQ2		0h	single bit and self clearing interrupt. writing a '1' to this bit will generate a pulse from DSS to MSS VIM line 61
0	MSSSWIRQ1		0h	single bit and self clearing interrupt. writing a '1' to this bit will generate a pulse from DSS to MSS VIM line 52

RADAR Subsystem

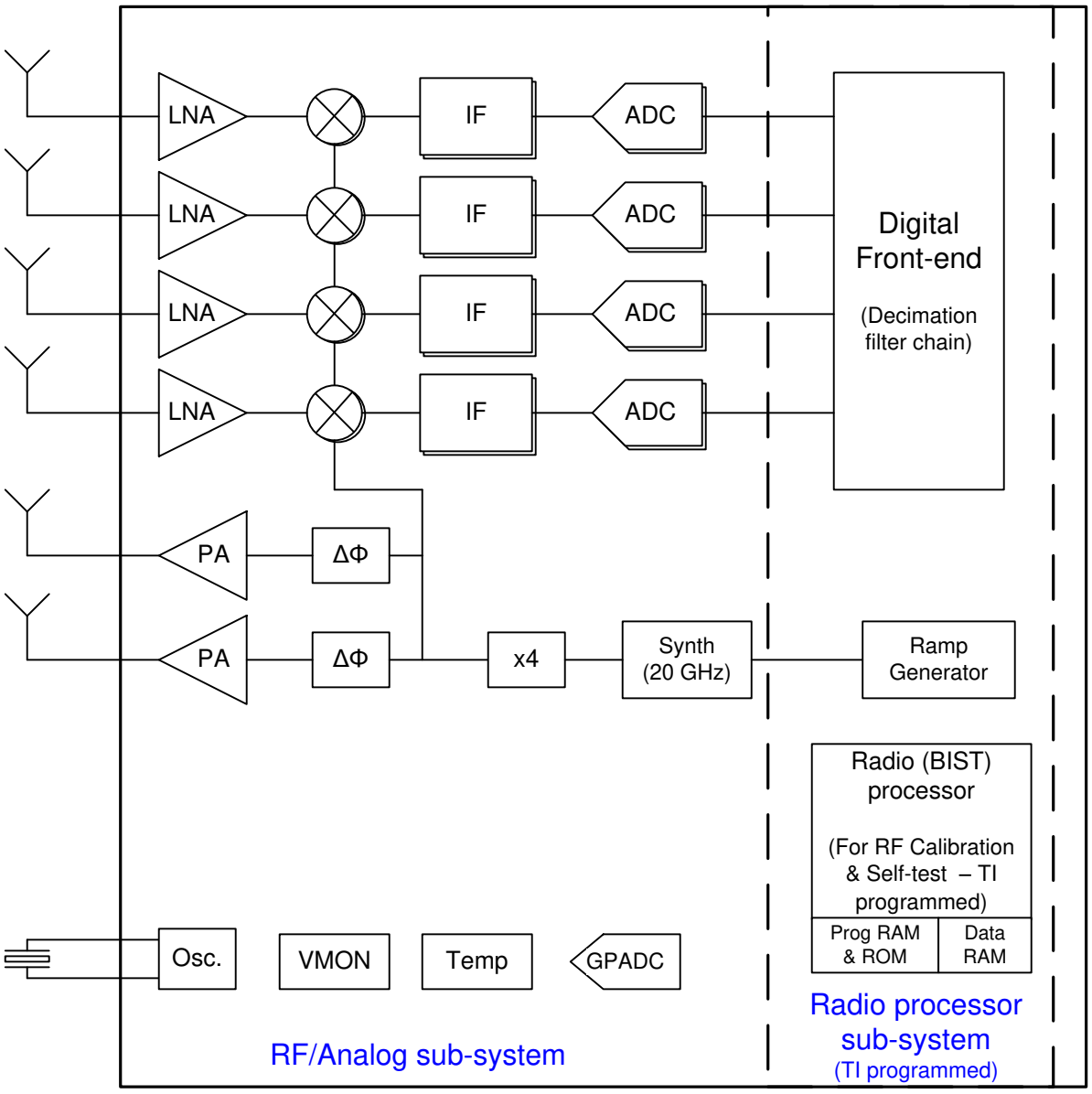
Topic **Page**

5.1 Block Diagram **1396**

5.1 Block Diagram

Figure 5-1 shows the RADAR subsystem block diagram.

Figure 5-1. RADAR Subsystem Block Diagram



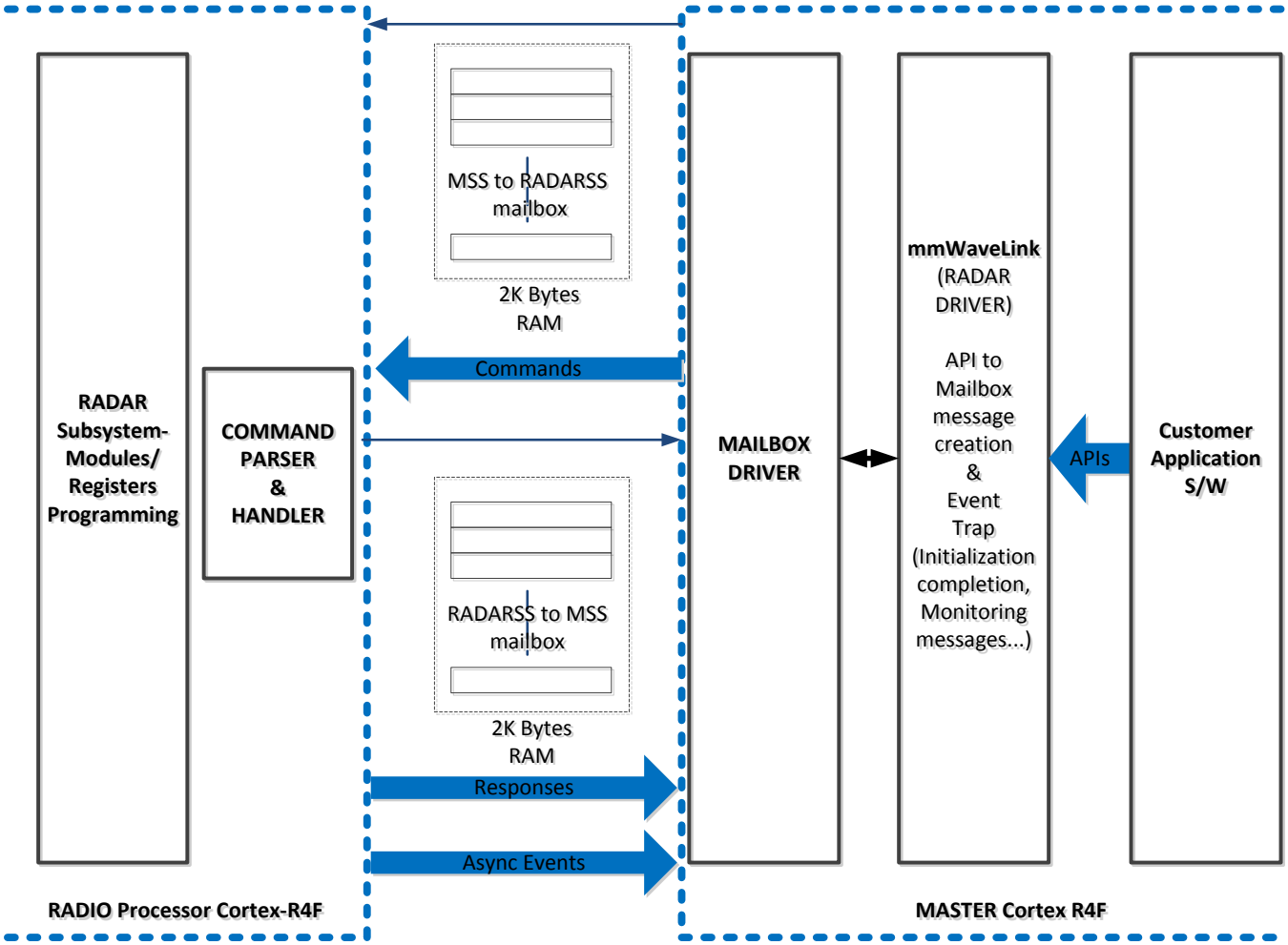
The RADAR subsystem consists of the RF/Analog subsystem and the radio processor subsystem.

The RF/analog subsystem implements the frequency-modulated continuous-wave transceiver system with RF and analog circuitry – namely, the synthesizer, PA, LNA, mixer, IF, and ADC. This subsystem also includes the crystal oscillator and temperature sensors. The two transmit channels can be operated up to a maximum of two at a time simultaneously. The four receive channels can all be operated simultaneously.

The radio processor subsystem includes the digital front-end, the ramp generator, and an internal processor for controlling and configuring the low-level RF/analog and ramp generator registers, based on well-defined API messages from the master subsystem. This radio processor is programmed by TI, and addresses both RF calibration needs and some basic self-test and monitoring functions (BIST); this processor is not available directly for customer use. The digital front-end filters and decimates the raw sigma-delta ADC output, and provides the final ADC data samples at a programmable sampling rate.

The programming of this RADAR subsystem is abstracted at a TI-defined API interface. Figure 5-2 below explains the high-level architecture and programming model.

Figure 5-2. Programming Model



The radar subsystem is accessed through a well-defined set of APIs. The application software running on the master Cortex-R4F processor, through the API call, invokes the mmWaveLink-Radar driver, which then converts these APIs into mailbox-based messages. When a message is written into the mailbox memory, an interrupt is raised to the radio processor, which then passes the message and executes the command conveyed through the mailbox message. A mailbox message read event is raised to the master CR4F to indicate the radio processor has read the message. For more details, refer to the API interface document.

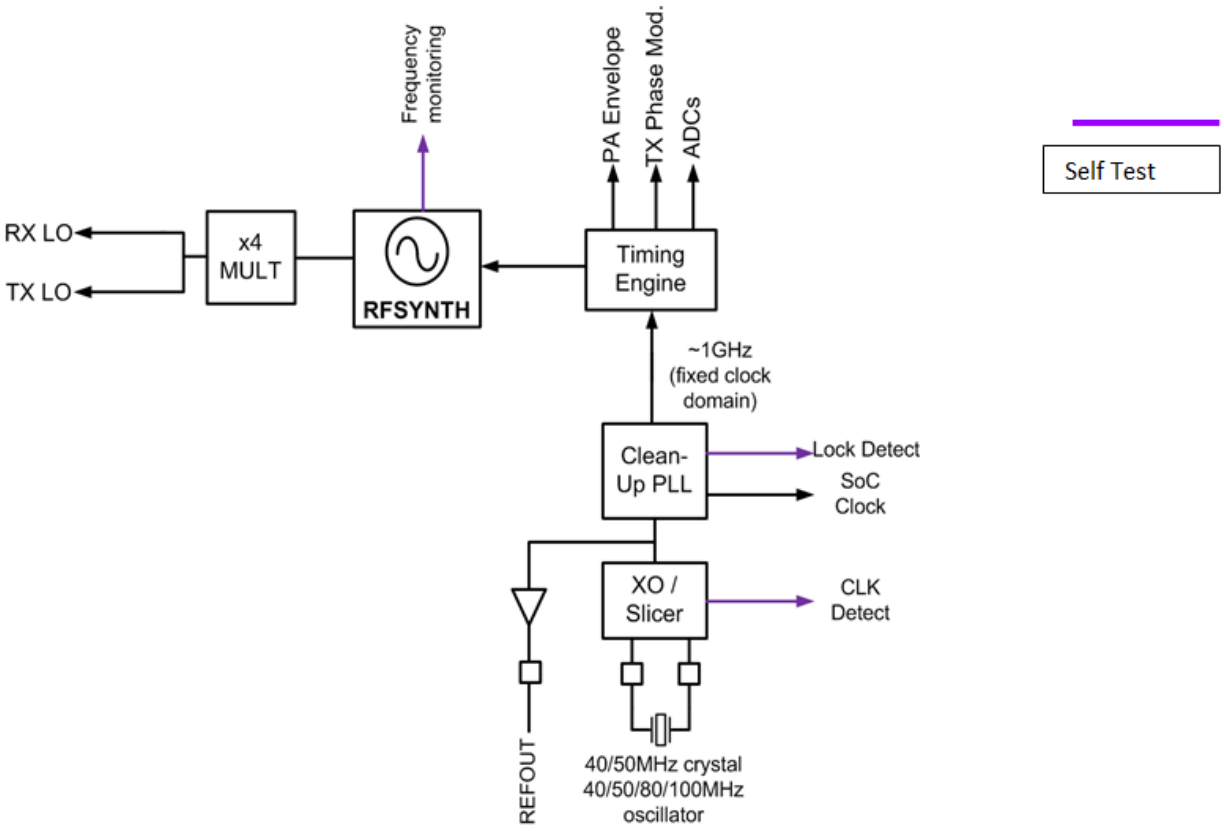
5.1.1 Clock Subsystem

The 16xx/14xx/18xx/68xx clock subsystem generates 76-71 (14xx/16xx/18xx) or 60-64 (68xx) GHz from an input reference of a 40-MHz crystal. The subsystem has a built-in oscillator circuit, followed by a clean-up PLL and an RF synthesizer circuit, as shown in Figure 5-3. The output of the RF synthesizer is then processed by an X4 multiplier to create the required frequency in the 76-71 (14xx/16xx/18xx) or 60-64 (68xx) GHz spectrum. The RF synthesizer output is modulated by the timing engine block to create the required waveforms for effective sensor operation. The timing engine is highly flexible and is programmed through the R4F-based radio controller subsystem. For details on waveforms and programming models, refer to the Radio Control API application note.

The clean-up PLL provides a reference clock for the host processor after system wakeup.

The clock subsystem has built-in mechanisms for detecting the presence of a crystal and monitoring the quality of the generated clock.

Figure 5-3. Clocking Subsystem

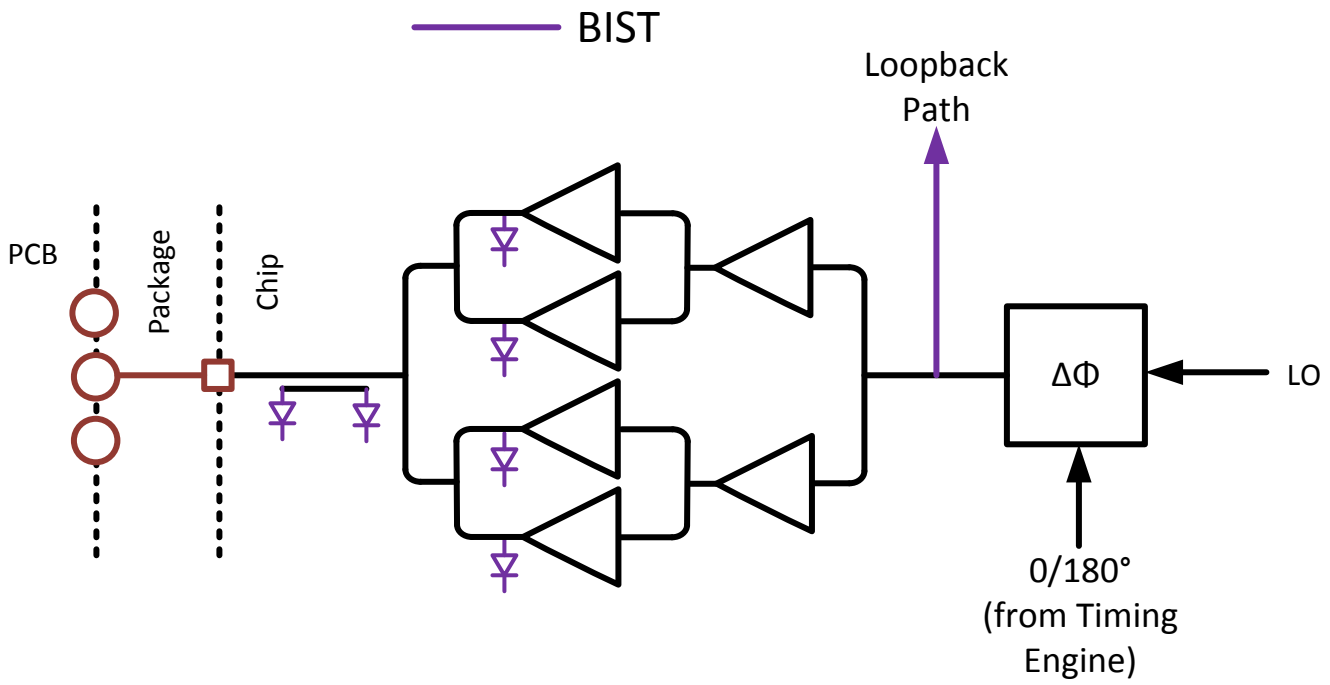


5.1.2 Transmit Subsystem

The transmit subsystem consists of three parallel transmit chains (other than the x16xx which has two), each with independent phase and amplitude control, as shown in Figure 5-4. Both transmit chains can be operational at the same time. The device supports binary phase modulation for MIMO radar and interference mitigation.

Each transmit chain is capable of delivering optimal power at the antenna port on the PCB, and supports programmable back-off for system optimization.

Figure 5-4. Transmit Subsystem



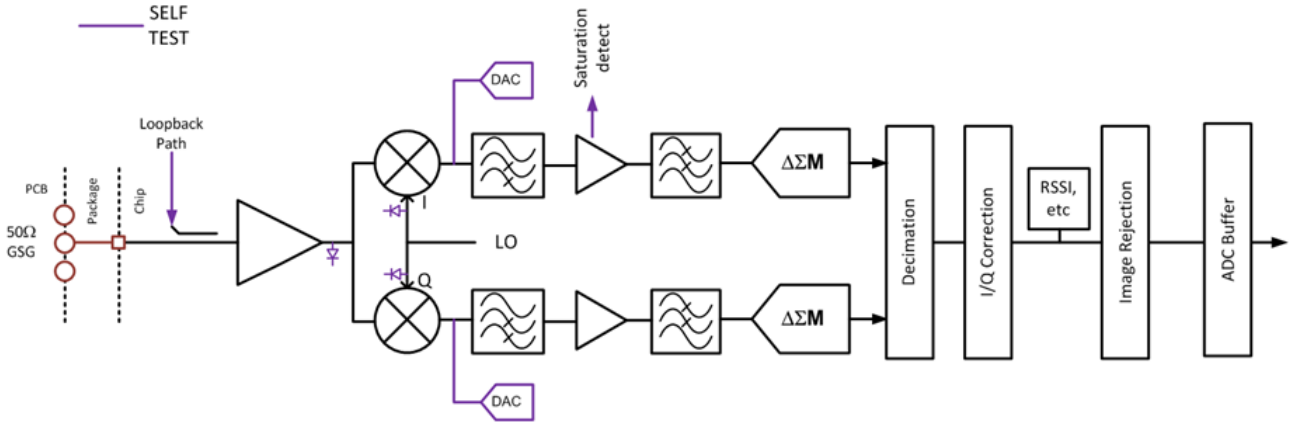
5.1.3 Receive Subsystem

The 16xx/14xx/18xx/68xx receive subsystem consists of four parallel channels, as shown in Figure 5-5. A single receive channel consists of an LNA, mixer, IF filtering, A2D conversion, and decimation. All four receive channels can be operational at the same time. An individual power down option is also available for system optimization.

Unlike conventional read-only receivers, the 16xx/14xx/18xx/68xx device supports a complex baseband architecture which uses quadrature mixer, dual IF, and ADC chains to provide complex I and Q outputs for each receiver channel. Refer to the Complex Receiver Baseband Application Note for further details.

The 16xx/14xx/18xx/68xx is targeted for fast chirp systems. The bandpass IF chain has configurable lower cutoff frequencies above 350 kHz, and can support bandwidths up to 15 MHz.

Figure 5-5. Receive Subsystem (Per Channel)



Master Subsystem Cortex-R4F

Topic **Page**

6.1 Introduction 1401
6.2 Cortex-R4F Subsystem 1401

6.1 Introduction

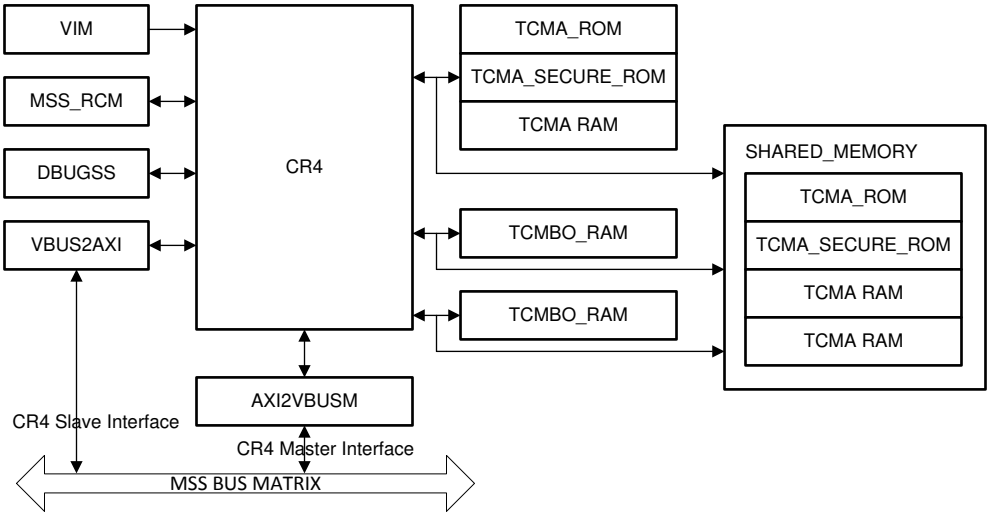
The master system includes an ARM Cortex R4F processor, clocked using a MSS_VCLK clock with a maximum operating frequency of 200 MHz. User applications executing on this processor control the overall operation of the device, including radar control through well-defined API messages, radar signal processing (assisted by the radar hardware accelerator), and peripherals for external interfaces.

Master (control) system enables autonomous operation of the device as a radar-on-a-chip sensor.

6.2 Cortex-R4F Subsystem

6.2.1 Cortex-R4F Integration

Figure 6-1. Cortex-R4F Integration



Features of Cortex R4F supported:

- No Icache/Dcache support
- TCMA, TCMB RAM/ROM supported
- Little Endianess only
- Memory protection supported

6.2.2 Tightly Coupled Memories

See the device-specific Integration chapter for

- Available dedicated TCMA RAM and TCMB RAM sizes for the Cortex-R4F processor in the master subsystem
- Total available L3 RAM in the device

A portion of L3 shared memory can be allotted to the Cortex-R4F as additional TCM, to further increase the TCMA and TCMB RAM. See the device-specific DSS_L3 chapter for details on sharing available L3 memory as MSS Coretex-R4F TCM.

6.2.3 Memory Module Hardware Initialization

The RAM memory can be initialized by using the dedicated auto-initialization hardware. The TCRAM wrapper initializes the entire memory when the auto-init is enabled for the RAM. All RAM data memory is initialized to zeros and the ECC memory is initialized to the correct ECC value for zeros, that is, 0x0C.

Direct Memory Access Controller (DMA)

The DMA controller is used to transfer data between two locations in the memory map in the background of CPU operations. Typically, the DMA is used to:

- Transfer blocks of data between external and internal data memories
- Restructure portions of internal data memory
- Continually service a peripheral

Topic **Page**

7.1 Module Operation **1403**
7.2 Control Registers and Control Packets **1422**
7.3 DMA Request Map **1508**

7.1 Module Operation

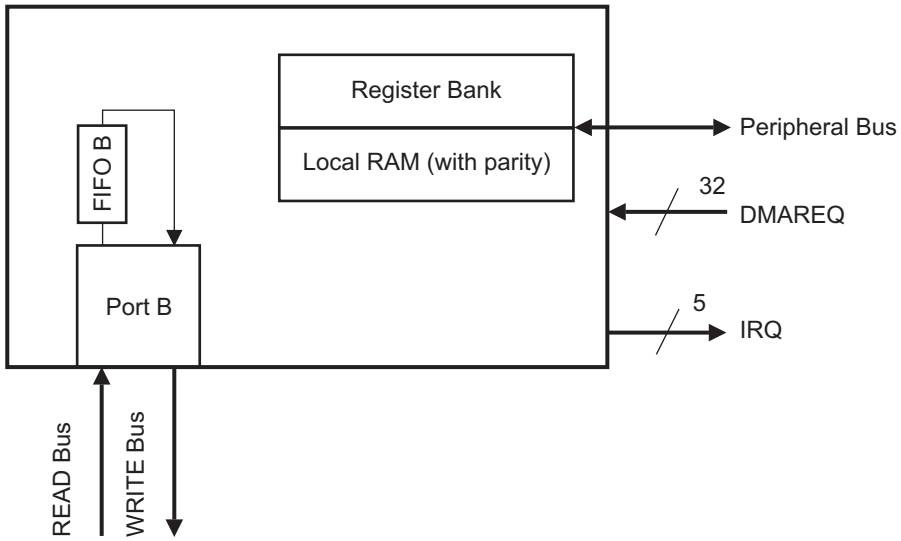
The DMA acts as an independent master in the platform architecture. All DMA memory and register accesses are performed in user mode. If the DMA writes to registers that are only accessible in privileged mode, the write will not be performed.

The DMA registers and its local RAM can only be accessed in privilege mode. Therefore, it is not possible for the DMA to reprogram itself.

7.1.1 Block Diagram

Figure 7-1 gives a detailed view of the DMA internal architecture. DMA data read and write access happens through Port B. FIFO B is 4 levels deep and 64-bits wide. 64 DMA requests go into the DMA that can trigger DMA transfers. Five interrupt request lines go out of the DMA to signal that a certain transfer status is reached. Register banks hold the memory mapped DMA configuration registers. Local RAM consists of DMA control packets and is secured by parity. All the programming and configuration of the DMA controller is done through the peripheral bus.

Figure 7-1. DMA Block Diagram



7.1.2 Memory Space

The DMA controller makes no distinction between program memory and data memory. The DMA controller can transfer to and from any space within the 4 gigabyte physical address map, by programming the absolute address for the source and destination in the control packet. Control packets store the transfer information such as source address, destination address, transfer count and control attributes for each channel.

7.1.3 DMA Data Access

The DMA controller refers to data in three levels of granularity:

- **Element:** Depending on the programmed data type, an 8-bit, 16-bit, 32-bit, or a 64-bit value. The type can be individually selected for the source (read) and destination (write). See Figure 7-2 and Figure 7-3 for an example of the use of elements. An element transfer cannot be interrupted.
- **Frame:** One or more elements to be transferred as a unit. A frame transfer can be interrupted between element transfers. See Figure 7-2 for an example.
- **Block:** One or more frames to be transferred as a unit. Each channel can transfer one block of data (once or multiple times). See Figure 7-3 for an example.

Figure 7-2. Example of a DMA Transfer Using Frame Trigger Source

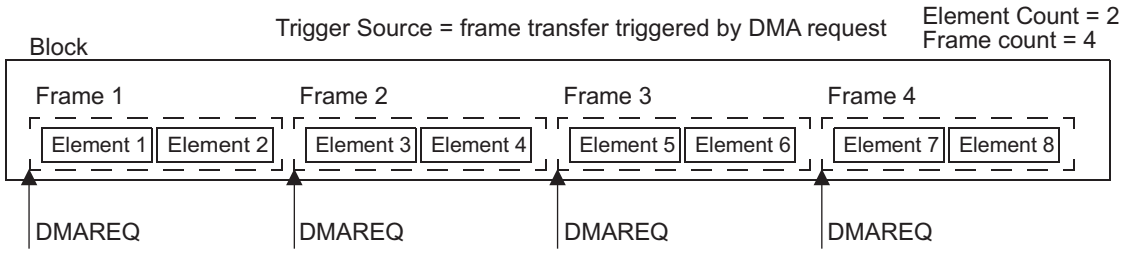
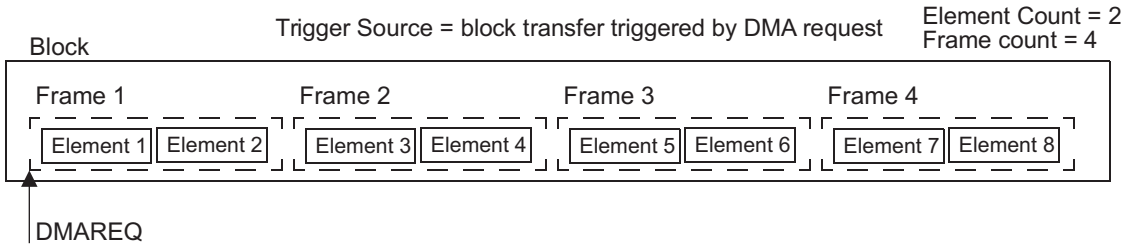


Figure 7-3. Example of a DMA Transfer Using Block Trigger Source



7.1.4 Addressing Modes

There are three addressing modes supported by the DMA controller that can be setup independent for the source and the destination address:

- Constant -- source and/or destination addresses do not change.
- Post incremented -- source and/or destination address are post-incremented by the element size.
- Indexed -- source and/or destination address is post-incremented as defined in the Element Index Offset Register (Section 7.2.2.5) and the Frame Index Offset Register (Section 7.2.2.6).

An unaligned address with respect to the element size is not supported.

7.1.5 DMA Channel Control Packets

There are a total of 16 control packets. Each control packet is associated with a channel in a fixed order. For example, control packet 0 stores channel information for channel 0. The DMA requests can be mapped to the individual channels as described in Section 7.1.8. The mapping scheme between DMA requests and channels is shown in Figure 7-4. Each control packet contains nine fields. The first six fields compose the primary control packet and are programmable during DMA setup. The last three fields compose working control packet and are only readable by the CPU. The working control packets are used to support auto-initiation. The organization of control packets is shown in Figure 7-5.

The primary control packet contains channel information such as source address, destination address, transfer count, element/frame index pointer and channel configuration. Source address, destination address and transfer count also have their respective working images. The three fields of working images compose a working control packet and are not accessible to the CPU in write access.

The first time a DMA channel is selected for a transaction, the following process occurs:

1. The primary control packet is first read by the DMA state machine.
2. Once the channel is arbitrated, the current source address, destination address and transfer count are then copied to their respective working images.
3. When the channel is serviced again by the DMA, the state machine will read both the primary control packet and the working control packet to continue the DMA transaction until the end of an entire block transfer.

When the same channel is requested again, the state machine will start again by reading only the primary control packet and then continue the same process described above. The user software need not set up control packets again because the contents of the primary control packet were never lost. The working images of the control packets are reducing the software overhead and interaction with the DMA module to a minimum.

NOTE: Changing the contents of a channel control packet will clear the corresponding pending bit () if the channel has a pending status. If the control packet of an active channel (as indicated in) is changed, then the channel will stop immediately at an arbitration boundary. When the same channel is triggered again, it will begin with the new control packet information.

7.1.5.1 Initial Source Address

This field stores the absolute 32-bit source address of the DMA transfer.

7.1.5.2 Initial Destination Address

This field stores the absolute 32-bit destination address of the DMA transfer.

Figure 7-4. DMA Request Mapping and Control Packet Organization

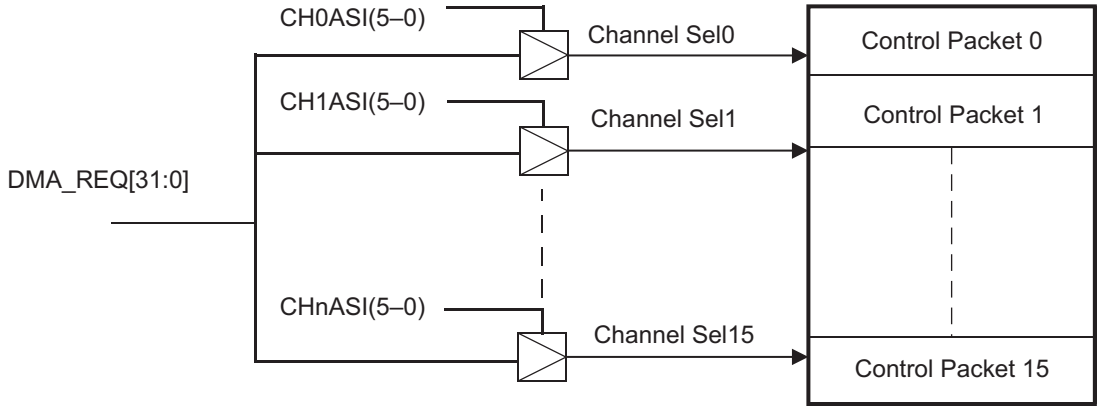
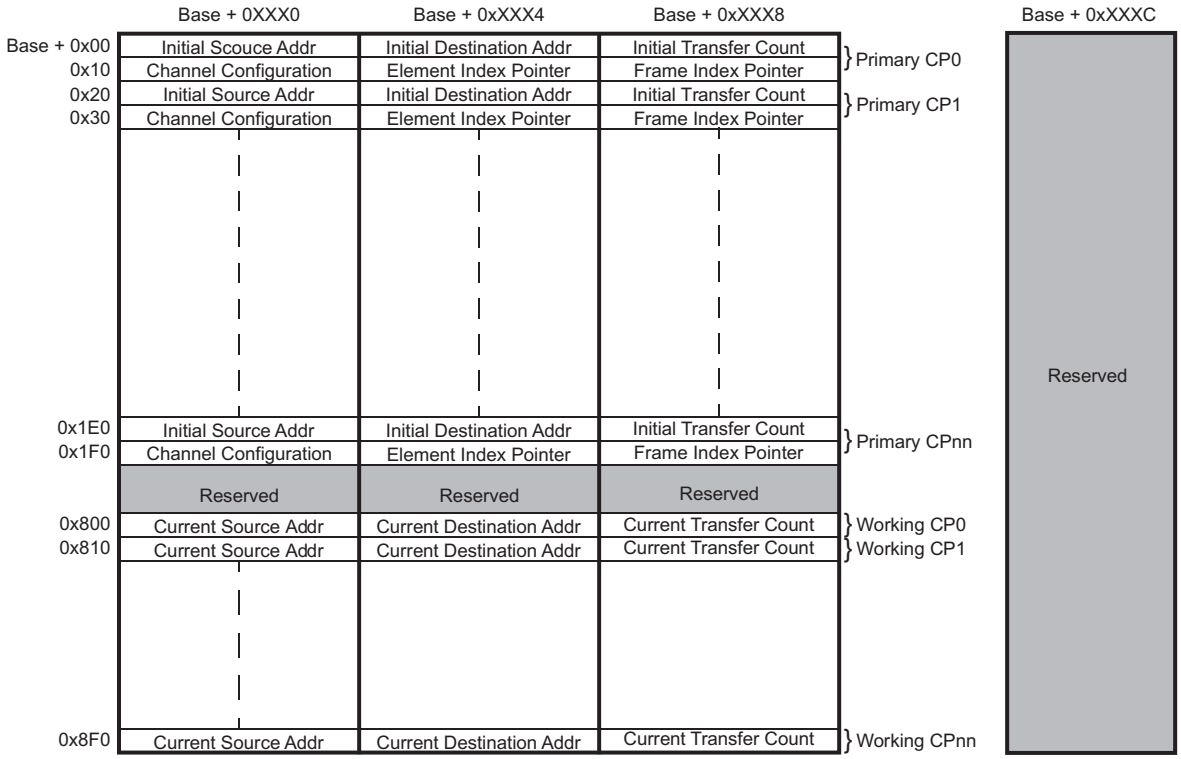


Figure 7-5. Control Packet Organization and Memory Map



7.1.5.3 Initial Transfer Count

The transfer count field is composed of two parts. The frame transfer count value and the element transfer count value. Each count value is 13 bits wide. As a Single Block transfer maximum of 512 Mbytes of data can be transferred. Element count and frame count are programmed according to the source data structure.

The total transfer size is calculated as:

$$T_{sz} = E_{rsz} \bullet E_{tc} \bullet F_{tc} \quad (1)$$

where

T_{sz} = Total Transfer Size

E_{rsz} = Read Element Size

E_{tc} = Element Transfer Count

F_{tc} = Frame Transfer Count

NOTE: A zero element count with a non-zero frame count or a non-zero element count with a zero frame count are all considered as zero total transfer count. No DMA transaction is initiated with any of the counters set to 0.

7.1.5.4 Channel Configuration Word

The channel configuration defines the following individual parameters

- Read element size
- Write element size
- Trigger type (frame or block)
- Addressing mode for source
- Addressing mode for destination
- Auto-initiation mode
- Next control packet to be triggered at control packet finish (Channel Chaining)

7.1.5.5 Element/Frame Index Pointer

There are 4 index pointers that allow the creation of different types of buffers in RAM and address registers in a structured manner: an element index pointer for source and destination and a frame index pointer for source and destination.

The element index pointer for source and/or destination defines the offset to be added after each element transfer to the source and/or destination address. The frame index pointer for source and/or destination defines the offset to be added to the source and/or destination address after the element count reaches zero. The element and frame index pointers must be defined in terms of the number of bytes of offset. The DMA controller does not adjust the element/frame index number according to the element size. An index of 2 means *increment the address by 2* and not by 16 when the element size is 64 bits.

7.1.5.6 Current Source Address

The current source address field contains the current working source address during a DMA transaction. The current source address is incremented during post increment addressing mode or indexing mode.

7.1.5.7 Current Destination Address

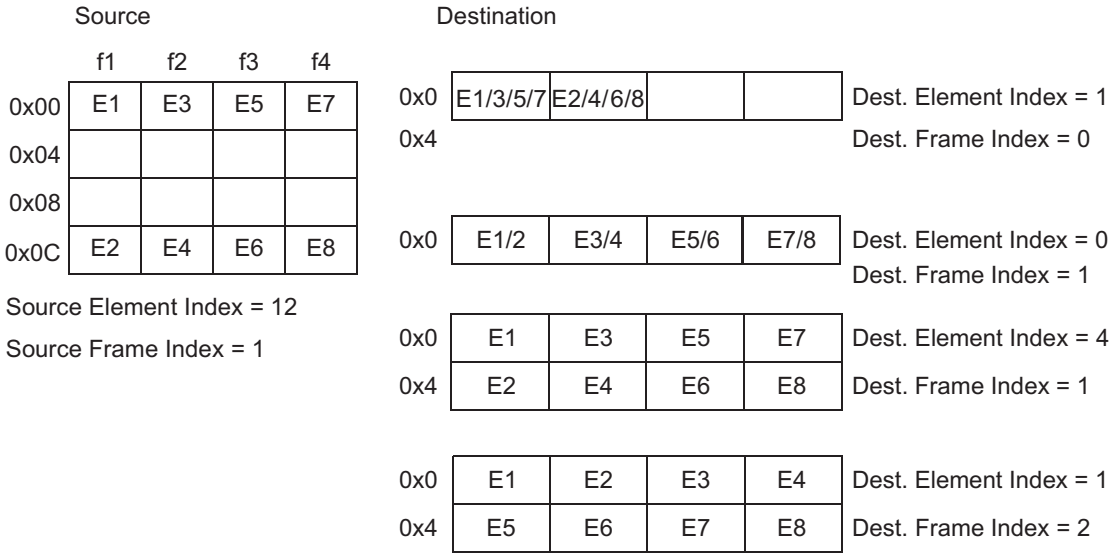
The current destination address field contains the current working destination address during a DMA transaction. The current destination address is incremented during post-increment addressing mode or indexing mode.

7.1.5.8 Current Transfer Count

The current transfer count stores the remaining number of elements to be transferred in a block. It is decremented by one for each element read from the source location.

Figure 7-6, Figure 7-7, and Figure 7-8 show some examples of DMA transfers.

Figure 7-6. DMA Transfer Example 1



The example assumes the following setup.

Read Element Size = 8 bit
Write Element Size = 8 bit
Element Count = 2
Frame Count = 4

Figure 7-7. DMA Indexing Example 1

	f1		f2		f3		f4	
0x0	E1		E5		E9		E13	
0x10	E2		E6		E10		E14	
0x20	E3		E7		E11		E15	
0x30	E4		E8		E12		E16	

Element Index = 16

Frame Index = 4

This example can be applied to either source or destination indexing and assumes the following setup.

Element Size = 16 bit
Element Count = 4
Frame Count = 4

Figure 7-8. DMA Indexing Example 2

0x0	E1	E4	E7	E10	E13	E16	E19	E22
0x20								
0x40	E2	E5	E8	E11	E14	E17	E20	E23
0x60								
0x80	E3	E6	E9	E12	E15	E18	E21	E23

Element Index = 64
Frame Index = 4

This example can be applied to either source or destination indexing and assumes the following setup.

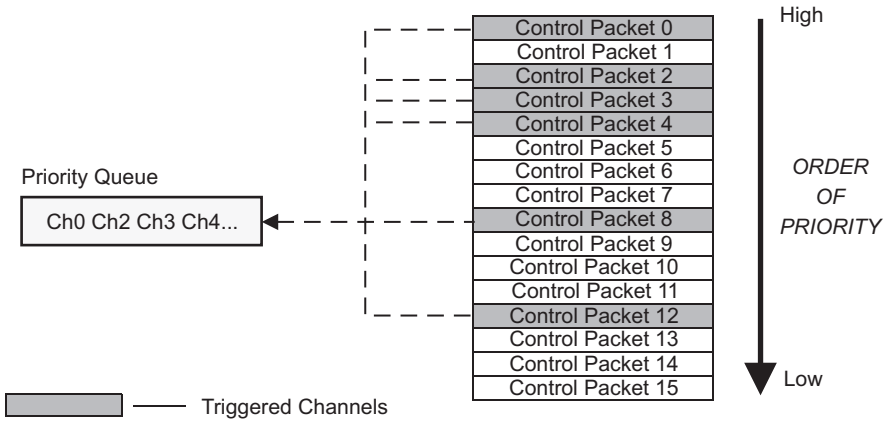
Element Size = 32 bit
Element Count = 3
Frame Count = 8

7.1.6 Priority Queue

User can assign channels in to priority queues to facilitate request handling during arbitration. The port has two priority queues: a high and a low priority queue. The queue can be configured to follow a fixed or rotating priority scheme. Fixed priority is such that the lower the channel number (Figure 7-9), the higher its priority. Rotating priority is based on a round-robin scheme. Initially, the priority list is sorted according to the fixed priority scheme. Channels assigned to the high priority queue are always serviced first according to the selected priority scheme before channels in the low priority queue are serviced. Table 7-1 describes how arbitration is performed according to different priority schemes.

NOTE: Since the DMA controller provides the capability to map any one of the 32 hardware DMA request lines to any channel, the numerical order of the hardware DMA request does not imply any priority. The priority of each hardware DMA request is programmed and determined by software.

Figure 7-9. Fixed Priority Scheme



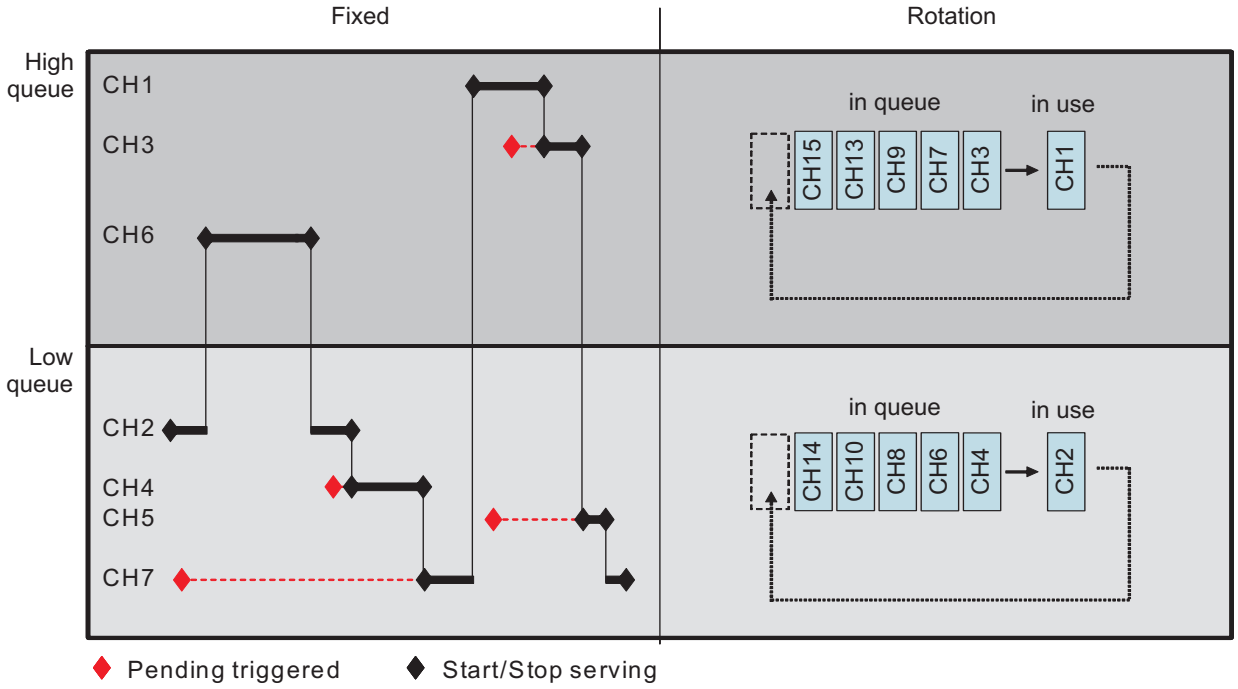
The above figure illustrates that by default Lower the channel number, higher the Priority.

Table 7-1. Arbitration According to Priority Queues and Priority Schemes

Queue	Priority Scheme	Remark
High priority	Fixed	Channels are serviced in an ascending order according to the channel number. The lower the channel number, the higher the priority. A channel will be arbitrated out whenever there is a higher pending channel. Otherwise a channel is completely serviced until its transfer count reaches zero before the next highest pending channel is serviced. When there is no pending channels left in high queue then the DMA switches to service low queue channels.
	Rotating	Channels are arbitrated by using the round-robin scheme. Arbitration is performed when the FIFO is empty. When there are no pending channels left in high queue then the DMA switches to service low queue channels.
Low priority	Fixed	Channels are serviced in an ascending order according to the channel number. The lower the channel number the higher the priority. A channel will be arbitrated out whenever there is a higher-priority pending channel. Otherwise a channel is completely serviced until its transfer count reaches zero, before the next highest pending channel is serviced. If there is a pending channel in the high-priority queue while DMA is servicing a low queue channel then DMA will switch back to service high queue channel after an arbitration boundary.
	Rotating	Channels are arbitrated by using round-robin scheme. Arbitration is performed when the FIFO is empty.

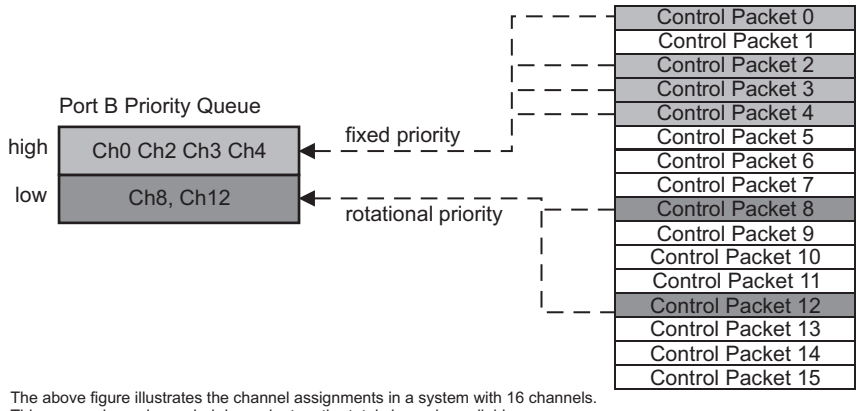
A Simple Priority Queues example in both Fixed and Rotation Scheme is shown in Figure 7-10.

Figure 7-10. Example of Priority Queues



For optimal system performance, the high priority channels should be put in fixed arbitration scheme and low priority channels in the rotating priority scheme as illustrated in Figure 7-11.

Figure 7-11. Example Channel Assignments



1 The above figure illustrates the channel assignments in a system with 16 channels. This approach can be scaled dependent on the total channels available.

7.1.7 Data Packing and Unpacking

The DMA controller automatically performs the necessary data packing and unpacking when the read element size differs from the write element size. Data packing is required when the read element size is smaller than the write element size; data unpacking is required when the read element size is larger than the write element size. When the read element size is equal to the write element size, no packing is performed during read, nor is any unpacking performed during write.

Figure 7-12 shows an example of data unpacking in which the DMA is used to transfer 128 transmit data elements to the MibSPI FIFO buffer. In this example, data unpacking is required because the read element size is 64 while the write element size is 16. The DMA first performs an 64-bit read from the source into its FIFO buffer. After the 64-bit data is read into the DMA FIFO buffer, it must unpack the data into four 16-bit data elements before writing out to the destination. Therefore the DMA would need to perform four 16-bit write operations to the destination.

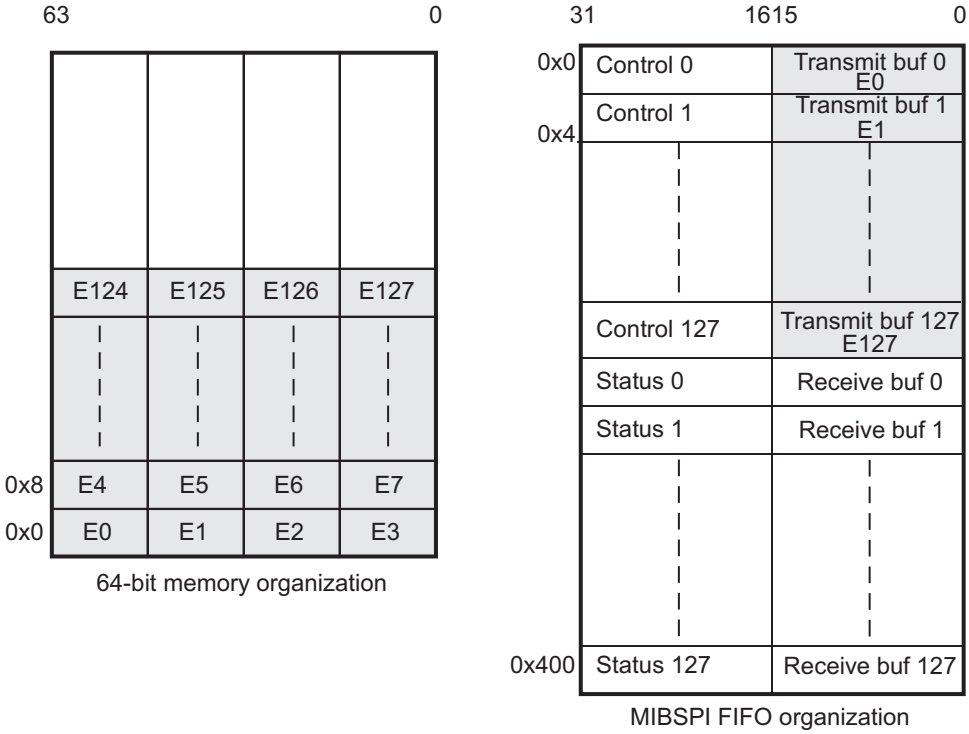
NOTE: In the example in Figure 7-12, to transmit data at the lower bits of the MibSPI, bits 15:0, the destination address should be incremented by a factor of 2.

NOTE: 1) The element Count (Section 7.2.2.3) refers only to the read element.

2) Data unpacking does not require the DMA request. Once the DMA request is received, data from Source is moved in to FIFO and unpacking happens until the FIFO is empty.

3) DMA assumes the destination is always ready and will perform write immediately. In case of data unpacking and Constant Addressing Mode write (Section 7.2.2.4 (1 - 0) = 0) the destination data will be overwritten by next data or next data might be skipped in case the destination has overflow protection (eg., SCITD register). User should configure DMA to avoid data unpacking if the Destination is configured as Constant Addressing Mode write to avoid data loss.

Figure 7-12. Example of DMA Data Unpacking



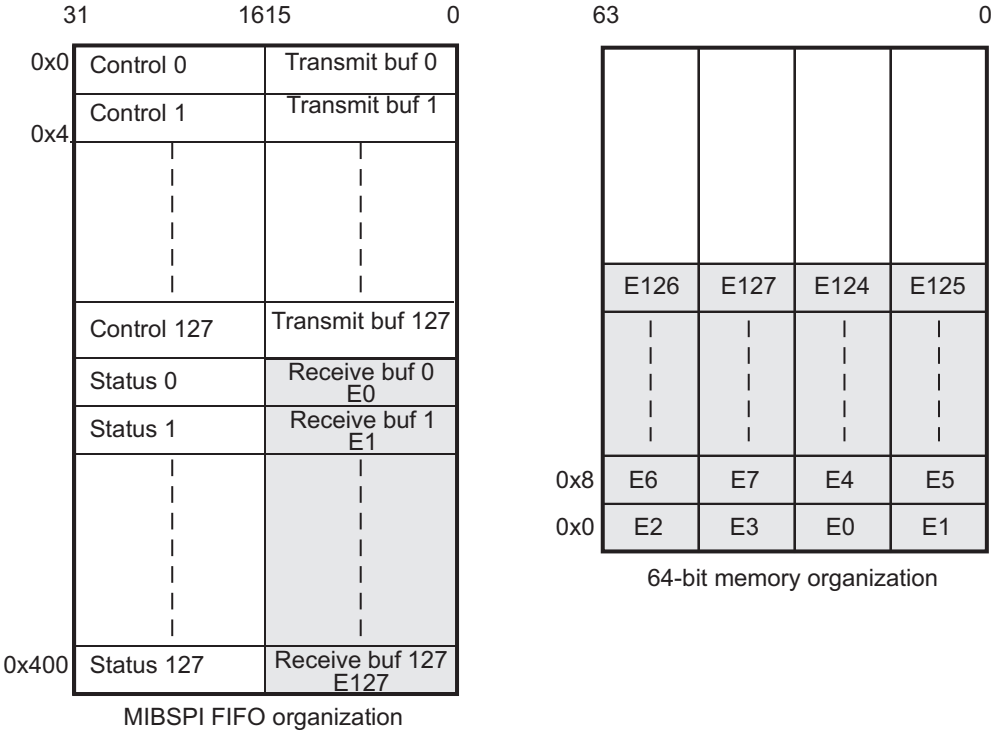
In this example, initialization of the MIBSPI FIFO is illustrated and assumes the following setup:

Read Element Size = 64 bit
Write Element Size = 16 bit
Element Count = 32
Frame Count = 1
Source Element Index = n/a, use post increment addressing mode
Source Frame Index = n/a, use post increment addressing mode
Destination Element Index = 4
Destination Frame Index = 0

When the read element size is smaller than the write element size, the DMA controller needs to perform data packing. The number of elements to pack is equal to the ratio between the write element size and read element size. In the example in Figure 7-13, the read element size is 16 bits and the write element size is 64 bits. The DMA controller would first pack the first four elements by performing four consecutive 16-bit read accesses of E0, E1, E2, and E3 into the first word of the DMA's internal FIFO. The DMA controller would then perform one single 64-bit write operation to transfer the data to the 64-bit destination memory.

Normally, the DMA controller carries out bus transactions on the bus according to the element size. For example, the DMA controller would perform a 16-bit read transaction if the read element size is programmed as 16 bits, or an 8-bit write transaction if the write element size is programmed as 8 bit. The exception is when the total transfer size is as defined in Equation 1 is not a multiple of the write element size.

Figure 7-13. Example of DMA Data Packing



In this example, a read of the MIBSPI FIFO is illustrated and assumes the following setup:

Read Element Size = 16 bit
Write Element Size = 64 bit
Element Count = 128
Frame Count = 1
Source Element Index = 4
Source Frame Index = 0
Destination Element Index = n/a, use post increment addressing mode
Destination Frame Index = n/a, use post increment addressing mode

For example, if the read element size is 8 bits, the element transfer count is equal to 9, and the write element size is 64 bit. The DMA controller would first perform eight 8-bit read transactions from the source. It would then perform a 64-bit write to the destination. When the same channel wins arbitration again, the DMA controller would first perform one 8-bit read from the source, followed by one 8-bit write to the destination, even though the write element size is 64 bit.

NOTE: Since peripherals are slower, it is advised to use data packing feature with caution for reading data from peripherals. Improper use might delay servicing other pending DMA channels.

7.1.8 DMA Request

There are three ways to start a DMA transfer:

- **Software request:** The transfer will be triggered by writing to SW Channel Enable Set and Status Register (). The software request can trigger either a block or a frame transfer depending on the setting of the TTYPE bit in the Channel Control Register (Section 7.2.2.4).
- **Hardware request**: The DMA controller can handle up to 32 DMA Request lines. A hardware request can trigger either a frame or a block transfer depending on the setting of the TTYPE bit in the Channel Control Register (Section 7.2.2.4).
- **Triggered by other control packet:** When a control packet finishes the programmed number of transfers it can trigger another channel to initiate its transfers.

Each time a DMA request is made, either one frame transfer or one block transfer can be chosen. An active DMA request signal will trigger a DMA transaction.

The DMA controller has a two-level buffer to capture HW requests per channel. When a HW request is generated and the channel is enabled, the corresponding bit in the DMA Status Register () is set. The pending register acts as a first-level buffer. Typically, a peripheral acting as a source of a transfer would initiate another request after its data registers have been read out by DMA, even though that data has not been completely transferred to the destination. If a second HW request is generated by the peripheral, the DMA controller has an extra request buffer to capture this second request and service it after the first request is complete.

NOTE: The DMA cannot capture more than three requests if its request buffers are already full. If any request occur during this moment DMA will discard it.

The DMA controller also supports a mix of hardware and software requests on the same channel. Note that such interchangeable usage may result into an out of sync for DMA channel and peripheral. The application needs to be careful as the DMA does not have a built-in mechanism to protect against this loss of synchronization.

If a software request is generated, the corresponding bit in the Channel Pending Register () is set accordingly. If the pending request is not completely serviced by the DMA and a hardware request is generated by a peripheral onto the same channel, the DMA will capture and recognize this hardware request into its request buffer.

NOTE: The DMA controller cannot recognize two software requests on the same channel if the first software request is still pending. If such request occur DMA will discard it. Therefore the user software should check the pending register before issuing a new software request.

The DMA module has 16 channels and up to 32 hardware DMA requests. The module contains DREQASIx registers that are used to map the DMA requests to the DMA channels. By default, channel 0 is mapped to request 0, channel 1 to request 1, and so on.

Some DMA requests have multiple sources, as shown in . The application must ensure that only one of these DMA request sources is enabled at any time.

7.1.9 Auto-Initiation

When Auto-initiation Mode (AIM) bit of Channel Control Register (Section 7.2.2.4) is enabled for a channel and the channel is triggered by a software request for a block transfer, the channel will restart again using the same channel information stored at the respective control packet after one block transfer is completed. In the case of Hardware Request, the channel needs to be retriggered each time after a block is complete even if auto-initiation is enabled.

7.1.10 Interrupts

Each channel can be configured to generate interrupts on several transfer conditions:

- Frame transfer complete (FTC) interrupt: an interrupt is issued after the last element of a frame has been transferred.
- Last frame transfer started (LFS) interrupt: an interrupt is issued before the first element of the last frame of a block transfer has started.
- First half of block complete (HBC) interrupt: an interrupt is issued if more than half of the block is transferred.
  - If the number of frames *n* is odd, then the HBC interrupt is generated at the end of the frame when *(n+1) / 2* number of frames are left in the block.
  - If the number of frames *n* is even, then the HBC interrupt is generated at the end of the frame after *n/2* number of frames are left in the block.
- Block transfer complete (BTC) interrupt: an interrupt is issued after the last element of the last frame has been transferred.
- External imprecise error on read: an interrupt can be issued when a bus error (Illegal transaction with ok response) is detected. The imprecise read error is connected to the ESM module.
- External imprecise error on write: an interrupt can be issued when a bus error (Illegal transaction with ok response) is detected. The imprecise write error is connected to the ESM module.
- Memory Protection Unit error (MPU): an interrupt is issued when the DMA detects that the access falls outside of a memory region programmed in the MPU registers of the DMA. The MPU interrupt is connected to the ESM module.
- Parity error (PAR): an interrupt is issued when the DMA detects a parity error when reading one of the control packets. The PAR interrupt is connected to the ESM module.

The DMA outputs 5 interrupt lines for control packet handling, a parity interrupt and a memory protection interrupt (Figure 7-14). Each type of transfer interrupt condition is grouped together. For example, all block-transfer complete interrupts that are routed to a port are combined (ORed). The channel that caused the interrupt is given in the corresponding interrupt channel offset register. Priority between interrupts among the same interrupt type is resolved by a fixed priority scheme. Priority between different interrupt types is resolved in the Vector Interrupt Manager. Figure 7-15 explains the Frame Transfer Complete Interrupt structure in detail.

NOTE: Each Channel Specific interrupts in DMA module are routed towards Group A or B to support two different CPUs individually. For devices with Single CPU or Dual CPU where both CPUs are running same code in delayed lock-step as safety feature:

Group A - Interrupts (FTC, LFS, HBC, and BTC) are routed to the ARM CPU.

Group B - Interrupts (FTC, LFS, HBC, and BTC) are not routed out.

User software should configure only Group A interrupts.

Figure 7-14. DMA Interrupts

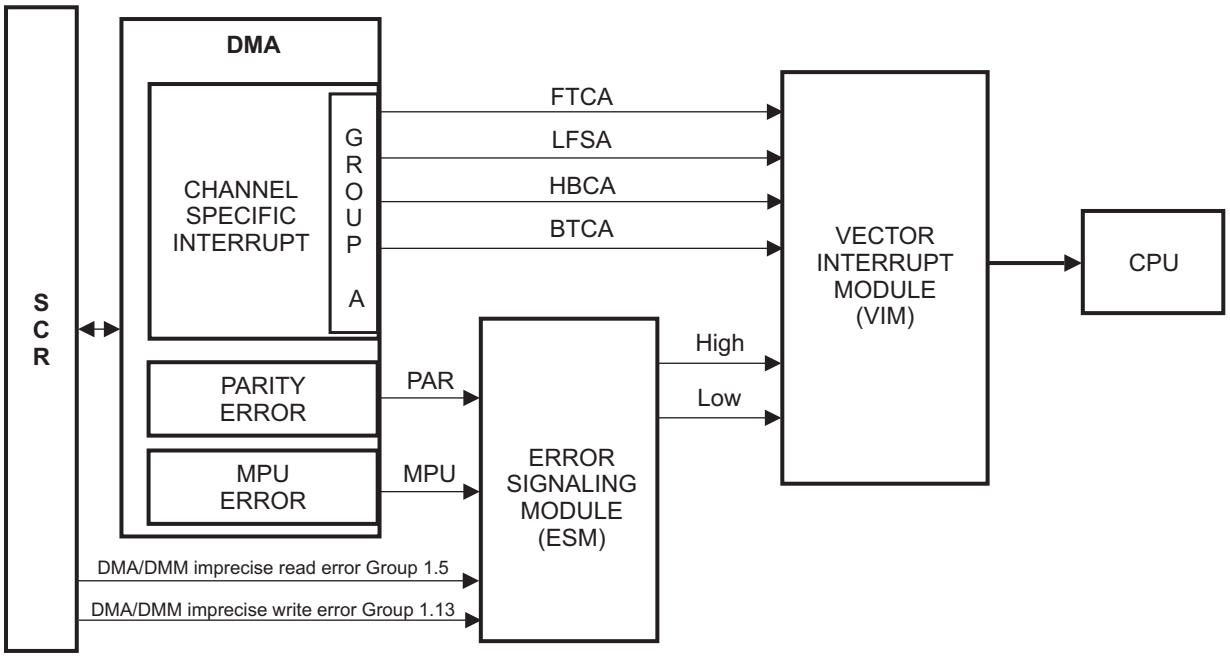
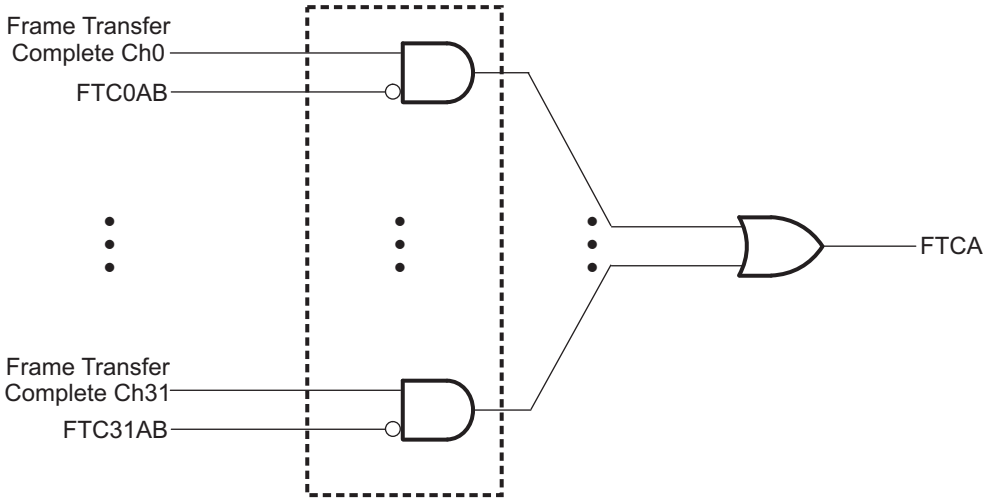


Figure 7-15. Detailed Interrupt Structure (Frame Transfer Complete Path)



This figure is applicable for the HBC, LFS, and BTC interrupt.

7.1.11 Debugging

The DMA supports four different behaviors in suspend mode. These behaviors can be configured by the user as per the application requirement.

- Immediate stop at a DMA channel arbitration boundary. Please refer to Table 7-2 and Table 7-3 for arbitration boundary definition.
- Finish current frame transfer and continue after suspend ends.
- Finish current block transfer and continue after suspend ends.
- Ignore the suspend. The DMA continues to be operational as in functional mode when debug mode is active.

When the DMA controller enters suspend mode, it continues to sample incoming hardware DMA requests, but the Channel Pending Register () is frozen from being updated. After the suspend ends, all new requests that were received during suspend mode are reflected in the Channel Pending Register ().

Except when the DMA controller is configured to ignore suspend mode, no channel arbitration is performed during suspend mode. The current channel under which suspend mode was entered will finish its entire frame or block-transfer after suspend mode ends, depending how the debug option was chosen.

To facilitate debugging, a Watch Point Register () and a Watch Mask Register () are used. The watch point register together with the watch mask register can be configured to watch for a unique address or a range of addresses. When the condition to watch is true, the DMA freezes its state and generates a debug request signal to the host CPU so the state of the DMA can be examined.

7.1.12 Power Management

The DMA offers two power-management modes: run and sleep. In run mode, the DMA is fully operational.

The sleep mode shuts down the DMA if no pending channels are waiting to be serviced. If a DMA request is received or a software request is generated by the user software, then the DMA wakes up immediately.

The sleep mode may be used to optimize the DMA module power consumption.

When the system module issues a global low power mode request, the DMA will respond to the system module with an acknowledge if no DMA requests are pending.

NOTE: When the DMA is in global low power mode, the clock is stopped and therefore it cannot detect any DMA request. The device must be woken up before a peripheral can generate a DMA request.

7.1.13 FIFO Buffer

DMA FIFO is 4 levels deep and 64-bit wide (can hold up to 4 x 64-bits of data). They are used for Data packing and unpacking.

The DMA FIFO has two states:

- EMPTY : The FIFO contains no data.
- FULL : The FIFO is filled or the element count has reached zero; the read operation has to be stopped.

DMA channels can only be switched when the FIFO is empty. This also implies that arbitration between channels is done when the FIFO is empty.

The FIFO buffer may be bypassed through the use of the bypass feature in the port control register; see Port Control Register () for register details. Writing 1 to this bit limits the FIFO depth to the size of one element. That means after one element is read the write out to the destination will start. This feature is particularly useful to minimize switching latency in-between channels. When bypass mode is enabled, the DMA will perform minimal bus cycles on AHB bus. In addition, the bypass feature allows arbitration between channels that can be carried out at a source element granularity.

However, it has to be considered that while in bypass mode, the DMA controller does not make optimal use of the bus bandwidth. Since the read and write element sizes can be different, then the number of read and write transactions will be different. Table 7-2 and Table 7-3 show a comparison between the number of read and write transactions performed by the DMA controller from one channel to another before arbitration in non-bypass and bypass mode.

Table 7-2. Maximum Number of DMA Transactions per Channel in Non-Bypass Mode

	Write Element Size	8 bit		16 bit		32 bit		64 bit	
Read Element Size	8 bit	4 read	4 write	4 read	2 write	4 read	1 write	8 read	1 write
	16 bit	2 read	4 write	4 read	4 write	4 read	2 write	4 read	1 write
	32 bit	1 read	4 write	2 read	4 write	4 read	4 write	4 read	2 write
	64 bit	1 read	8 write	1 read	4 write	2 read	4 write	4 read	4 write

Table 7-3. Maximum Number of DMA Transactions per Channel in Bypass Mode

	Write Element Size	8 bit		16 bit		32 bit		64 bit	
Read Element Size	8 bit	1 read	1 write	2 read	1 write	4 read	1 write	8 read	1 write
	16 bit	1 read	2 write	1 read	1 write	2 read	1 write	4 read	1 write
	32 bit	1 read	4 write	1 read	2 write	1 read	1 write	2 read	1 write
	64 bit	1 read	8 write	1 read	4 write	1 read	2 write	1 read	1 write

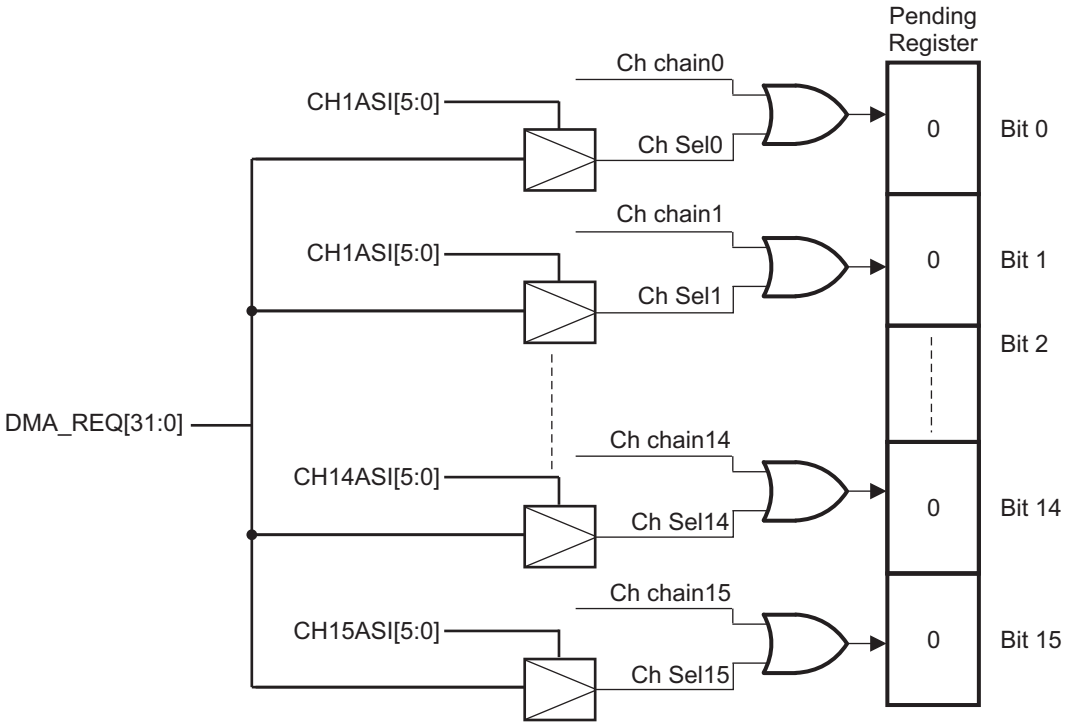
7.1.14 Channel Chaining

Channel chaining is used to trigger a single or multiple channels with out an external DMA request. This is possible by chaining one control packet to other. Chain[5:0] field of the Channel Control Register (Section 7.2.2.4) is used to program the chaining control packet. Chained control packets follow arbitration rules within the pending register. For example if CH1, CH2, CH4, CH5 are triggered together and CH3 is chained with CH1. The order of channels serviced in spite of chaining will be CH1 -> CH2 -> CH3 -> CH4 -> CH5.

In order to setup up channel chain feature, the Channel Control Register (Section 7.2.2.4) needs to be enabled for all chained channels before triggering first DMA request.

Figure 7-16 illustrates how internally chained request is generated after completing the required transfers and stored in pending register. In this example CH1 is Chained to CH0. When CH0 is triggered CH1 is captured as pending in the Channel Pending Register () even when it is not triggered.

Figure 7-16. Example of Channel Chaining



7.1.15 Memory Protection

The DMA controller is capable of access to the full address range of the device. The protection mechanism allows the protection of up to four memory regions to restrict accesses to those address ranges. This will allow the application to protect critical application data from unintentionally being accessed by the DMA controller.

7.1.15.1 Protection Mechanism

The memory protection mechanism consists of the access privilege for a given memory region, the start and end address for the region, and notification of an access violation for the protected region.

Each region to be protected is configured by software by writing the start address and end address for each region into the DMA Memory Protection Registers, DMAMPRxS and DMAMPRxE. The definition of these registers can be found starting at . Any region in the valid address space can be protected from inappropriate accesses.

The access privileges can be set to one of four permission settings as shown below:

- Full access
- Read only access
- Write only access
- No access

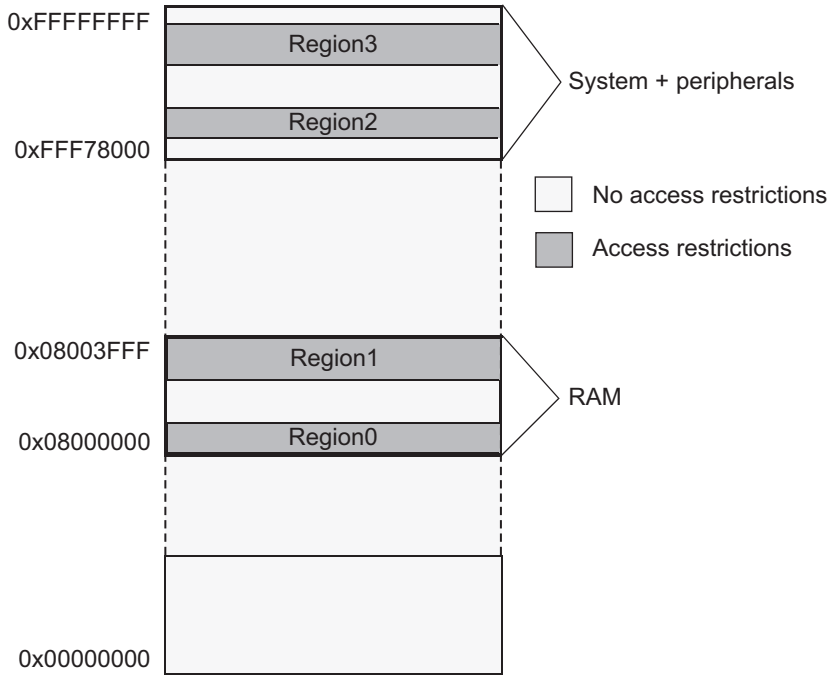
The permissions for a given region are selected by writing the appropriate values in the DMA Memory Protection Control Register ().

NOTE: If the regions defined by the start and end addresses overlap, the region defined first in the register space determines the access privilege. For example, if region 0 and region 1 overlap, the access permissions defined for region 0 will take precedence since region 0 registers are before region 1.

In a case where a memory protection violation occurs, a flag will be set and an interrupt will be generated, if interrupts are enabled. The DMA Memory Protection Status Register () contains the status flags for the memory protection mechanism, and the DMA Memory Protection Control Register () contains the interrupt enable bits. Upon detection of the memory protection violation, the DMA Channel that caused the violation will be stopped and the next available DMA channel will be serviced.

Figure 7-17 Illustrates a protection mechanism.

Figure 7-17. Example of Protection Mechanism



7.1.16 Parity Checking

Parity checking is implemented using parity on a per-byte basis for DMA Control Packets in the RAM. Checking for even or odd parity can be programmed by a 4-bit key located in the system module that controls the parity configuration on a global basis. This ensures that all modules using parity are acting in the same manner. The default setup after reset is odd parity.

In addition, parity checking can be enabled and disabled within the module by a 4-bit key. The key is located in the Parity Control Register ().

During a read access, regardless if it was read by the DMA state machine or another master (CPU), the parity is calculated based on the data read from the RAM and compared with the good parity value stored in the parity bits. If any word fails the parity check, then a parity error interrupt is generated. The address that generated the error is detected and is captured for host system debugging in the DMA Parity Error Address Register (). The address is frozen from being updated until it is read by the bus master.

Additional error handling is dependent on the requestor.

- DMA reading from a control packet RAM: The transmission requested by DMA request will not take place.
- CPU reading from the control packet RAM: The data will be retrieved by the CPU and a parity error interrupt will be generated.

In both cases, the control packet will be left active or the DMA will be switched off dependent on the ERRA bit in the Parity Control Register ().

7.1.17 Parity Testing

The parity RAM is accessible to allow manually inserting faults so that the parity checking feature can be tested. Test mode is entered by asserting the TEST bit in the Parity Control Register (). Once the bit is set, the parity bits are mapped to the control packet RAM starting address A00h.

NOTE: When in test mode, no parity checking will be done when reading from parity memory, but parity checking will be performed on the normal memory.

Each byte in Control Packet RAM has its own parity bit in the Control Packet Parity RAM as shown in Table 7-4, Table 7-5, and Table 7-6. P0 is the parity bit for byte 0, P1 is the parity bit for byte 1 and so on.

Each byte in the control packet RAM has its own parity bit in the control packet parity RAM as shown in Table 7-4 and Table 7-5.

Table 7-4. Control Packet RAM

Bit	31 24	23 16	15 8	7 0
Word0	Byte 0	Byte 1	Byte 2	Byte 3
Word1	Byte 4	Byte 5	Byte 6	Byte 7
Word2	Byte 8	Byte 9	Byte 10	Byte 11
Word3	Byte 12	Byte 13	Byte 14	Byte 15

Table 7-5. Control Packet RAM

Bit	127 96	95 64	63 32	31 0
	Word 3	Word 2	Word 1	Word 0

Table 7-6. Parity RAM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0

7.1.18 Initializing RAM with Parity

After power up, the RAM content including the parity bit cannot be guaranteed. To avoid parity failures when reading RAM, the RAM has to be initialized. The RAM can be initialized by writing known values into it. When the known value is written, the corresponding parity bit will be automatically calculated and updated.

Another possibility to initialize the memory is to follow the Auto-Initialization of On-Chip SRAM Modules subsection in the *Architecture* chapter. The RAM will be initialized to 0. Depending on the even/odd parity selection, the parity bit will be calculated accordingly.

To allow for parity calculation during initialization, the parity functionality has to be enabled as discussed in Section 7.1.16.

7.2 Control Registers and Control Packets

The master subsystem has two DMA modules, DMA1 and DMA2. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved. The DMA control registers are summarized in Table 7-7. The control packets are summarized in Table 7-8. All registers and control packets are accessible in 8,16, and 32 bit.

NOTE: The register definitions are given for a full DMA module configuration (32 channels, 64 requests, 2 Ports, Dual CPU support). Writes and Reads of bits pertaining to features not included in the DMA implementation as defined in the device-specific data manual are possible without error; however, they will have no affect on device operation.

7.2.1 DMA Control Registers

Table 7-7 lists the memory-mapped DMA registers. All register offset addresses not listed in Table 7-7 should be considered as reserved locations and the register contents should not be modified.

Table 7-7. DMA Control Registers

Offset	Acronym	Register Name	Section
0h	GCTRL	GLOBAL CONTROL Register	Section 7.2.1.1
4h	PEND	CHANNEL PENDING Register	Section 7.2.1.2
8h	FBREG	FALL BACK Register	Section 7.2.1.3
Ch	DMASTAT	DMA STATUS Register	Section 7.2.1.4
14h	HWCHENAS	HW CHANNEL ENABLE SET Register	Section 7.2.1.5
1Ch	HWCHENAR	HW CHANNEL ENABLE RESET Register	Section 7.2.1.6
24h	SWCHENAS	SW CHANNEL ENABLE SET Register	Section 7.2.1.7
2Ch	SWCHENAR	SW CHANNEL ENABLE RESET Register	Section 7.2.1.8
34h	CHPRIOS	CHANNEL PRIORITY SET Register	Section 7.2.1.9
3Ch	CHPRIOR	CHANNEL PRIORITY RESET Register	Section 7.2.1.10
44h	GCHIENAS	GLOBAL CHANNEL INTERRUPT ENABLE SET Register	Section 7.2.1.11
4Ch	GCHIENAR	GLOBAL CHANNEL INTERRUPT ENABLE RESET Register	Section 7.2.1.12
54h	DREQASI0	DMA REQUEST ASSIGNMENT Register 0	Section 7.2.1.13
58h	DREQASI1	DMA REQUEST ASSIGNMENT Register 1	Section 7.2.1.14
5Ch	DREQASI2	DMA REQUEST ASSIGNMENT Register 2	Section 7.2.1.15
60h	DREQASI3	DMA REQUEST ASSIGNMENT Register 3	Section 7.2.1.16
64h	DREQASI4	DMA REQUEST ASSIGNMENT Register 4	Section 7.2.1.17
68h	DREQASI5	DMA REQUEST ASSIGNMENT Register 5	Section 7.2.1.18
6Ch	DREQASI6	DMA REQUEST ASSIGNMENT Register 6	Section 7.2.1.19
70h	DREQASI7	DMA REQUEST ASSIGNMENT Register 7	Section 7.2.1.20
94h	PAR0	PORT ASSIGNMENT Register 0	Section 7.2.1.21
98h	PAR1	PORT ASSIGNMENT Register 1	Section 7.2.1.22
9Ch	PAR2	PORT ASSIGNMENT Register 2	Section 7.2.1.23
A0h	PAR3	PORT ASSIGNMENT Register 3	Section 7.2.1.24
B4h	FTCMAP	FTC INTERRUPT MAPPING Register	Section 7.2.1.25
BCh	LFSMAP	LFS INTERRUPT MAPPING Register	Section 7.2.1.26
C4h	HBCMAP	HBC INTERRUPT MAPPING Register	Section 7.2.1.27
CCh	BTCMAP	BTC INTERRUPT MAPPING Register	Section 7.2.1.28
D4h	BERMAP	BER INTERRUPT MAPPING Register	Section 7.2.1.29
DCh	FTCINTENAS	FTC INTERRUPT ENABLE SET Register	Section 7.2.1.30
E4h	FTCINTENAR	FTC INTERRUPT ENABLE RESET Register	Section 7.2.1.31
ECh	LFSINTENAS	LFS INTERRUPT ENABLE SET Register	Section 7.2.1.32
F4h	LFSINTENAR	LFS INTERRUPT ENABLE RESET Register	Section 7.2.1.33
FCh	HBCINTENAS	HBC INTERRUPT ENABLE SET Register	Section 7.2.1.34
104h	HBCINTENAR	HBC INTERRUPT ENABLE RESET	Section 7.2.1.35
10Ch	BTCINTENAS	BTC INTERRUPT ENABLE SET Register	Section 7.2.1.36
114h	BTCINTENAR	BTC INTERRUPT ENABLE RESET Register	Section 7.2.1.37
11Ch	GINTFLAG	GLOBAL INTERRUPT FLAG Register	Section 7.2.1.38
124h	FTCFLAG	FTC INTERRUPT FLAG Register	Section 7.2.1.39
12Ch	LFSFLAG	LFS INTERRUPT FLAG Register	Section 7.2.1.40
134h	HBCFLAG	HBC INTERRUPT FLAG Register	Section 7.2.1.41
13Ch	BTCFLAG	BTC INTERRUPT FLAG Register	Section 7.2.1.42

Table 7-7. DMA Control Registers (continued)

Offset	Acronym	Register Name	Section
144h	BERFLAG	BER INTERRUPT FLAG Register	Section 7.2.1.43
14Ch	FTCAOFFSET	FTCA INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.44
150h	LFSAOFFSET	LFSA INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.45
154h	HBCAOFFSET	HBCA INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.46
158h	BTCAOFFSET	BTCA INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.47
15Ch	BERAOFFSET	BERA INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.48
160h	FTCBOFFSET	FTCB INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.49
164h	LFSBOFFSET	LFSB INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.50
168h	HBCBOFFSET	HBCB INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.51
16Ch	BTCBOFFSET	BTCB INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.52
170h	BERBOFFSET	BERB INTERRUPT CHANNEL OFFSET Register	Section 7.2.1.53
178h	PTCRL	PORT CONTROL Register	Section 7.2.1.54
17Ch	RTCTRL	RAM TEST CONTROL Register	Section 7.2.1.55
180h	DCTRL	DEBUG CONTROL Register	Section 7.2.1.56
184h	WPR	WATCH POINT Register	Section 7.2.1.57
188h	WMR	WATCH MASK Register	Section 7.2.1.58
18Ch	PAACSADDR	PORT A ACTIVE CHANNEL SOURCE ADDRESS	Section 7.2.1.59
190h	PAACDADDR	PORT A ACTIVE CHANNEL DESTINATION ADDRESS	Section 7.2.1.60
194h	PAACTC	PORT A ACTIVE CHANNEL TRANSFER COUNT	Section 7.2.1.61
198h	PBACSADDR	PORT B ACTIVE CHANNEL SOURCE ADDRESS	Section 7.2.1.62
19Ch	PBACDADDR	PORT B ACTIVE CHANNEL DESTINATION ADDRESS	Section 7.2.1.63
1A0h	PBACTC	PORT B ACTIVE CHANNEL TRANSFER COUNT	Section 7.2.1.64
1A8h	DMAPCR	PARITY CONTROL REGISTER	Section 7.2.1.65
1ACh	DMAPAR	PARITY ERROR ADDRESS	Section 7.2.1.66
1B0h	DMAMPCTRL	MEMORY PROTECTION CONTROL REGISTER	Section 7.2.1.67
1B4h	DMAMPST	MEMORY PROTECTION STATUS REGISTER	Section 7.2.1.68
1B8h	DMAMPR0S	MEMORY PROTECTION REGION 0 START ADDRESS	Section 7.2.1.69
1BCh	DMAMPR0E	MEMORY PROTECTION REGION 0 END ADDRESS	Section 7.2.1.70
1C0h	DMAMPR1S	MEMORY PROTECTION REGION 1 START ADDRESS	Section 7.2.1.71
1C4h	DMAMPR1E	MEMORY PROTECTION REGION 1 END ADDRESS	Section 7.2.1.72
1C8h	DMAMPR2S	MEMORY PROTECTION REGION 2 START ADDRESS	Section 7.2.1.73
1CCh	DMAMPR2E	MEMORY PROTECTION REGION 2 END ADDRESS	Section 7.2.1.74
1D0h	DMAMPR3S	MEMORY PROTECTION REGION 3 START ADDRESS	Section 7.2.1.75
1D4h	DMAMPR3E	MEMORY PROTECTION REGION 3 END ADDRESS	Section 7.2.1.76

Complex bit access types are encoded to fit into small table cells. Table 7-8 shows the codes that are used for access types in this section.

Table 7-8. MSS_DMA_REG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

7.2.1.1 GCTRL Register (Offset = 0h) [reset = 0h]

GCTRL is shown in Figure 7-18 and described in Table 7-9.

Return to Summary Table.

Sets Power-down mode, debug mode...

Figure 7-18. GCTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA EN
R/W-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED	BUS BUSY	RESERVED				DEBUG MODE	
R/W-0h	R-0h	R/W-0h				R/W-0h	
7	6	5	4	3	2	1	0
RESERVED							DMARES
R/W-0h							R/W-0h

Table 7-9. GCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	Reserved
16	DMA EN	R/W	0h	DMA Enable bit
15	RESERVED	R/W	0h	Reserved
14	BUS BUSY	R	0h	BUS BUSY This bit indicates status of DMA external AHB bus status. 0 DMA's external bus is not busy in data transfers. 1 DMA's external bus is busy in data transfers
13-10	RESERVED	R/W	0h	Reserved
9-8	DEBUG MODE	R/W	0h	DEBUG MODE 11 immediate stop at an DMA arbitration boundary and continue after suspend 10 finish current frame transfer 01 finish current block transfer 00 ignore suspend
7-1	RESERVED	R/W	0h	Reserved
0	DMARES	R/W	0h	DMA Software Reset Writing a 1 to this bit resets the DMA state machine and all control registers. Control packets are not reset when DMA software reset is active. Writing a zero disable SW reset.

7.2.1.2 PEND Register (Offset = 4h) [reset = 0h]

PEND is shown in Figure 7-19 and described in Table 7-10.

Return to Summary Table.

Channel pending register

Figure 7-19. PEND Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEND																															
R/W-0h																															

Table 7-10. PEND Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PEND	R/W	0h	Channel Pending Register Writing has no effect. Reading from PEND gives the channel pending information no matter if the channel was initiated by SW or HW. Once set, it remains set even if the corresponding channel is disabled via HWCHENA or SWCHENA. The pending bit is automatically cleared for the following conditions: 1)At the end of a frame or a block transfer depending on how the channel is triggered as programmed in the TTYPE bit field of Section 1.6.3.4, "CHANNEL CONTROL Register (CHCTRL)" . 2)The control packet is modified after the pending bit is set. 3)An AHB bus error occurs. 1 the associated channel is pending and is waiting for service. 0 the associated channel is inactive.

7.2.1.3 FBREG Register (Offset = 8h) [reset = 505h]

FBREG is shown in Figure 7-20 and described in Table 7-11.

Return to Summary Table.

Fall back register for EMC

Figure 7-20. FBREG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FSM FB				RESERVED				VBUSP FB			
R-0h				R/W-5h				R-0h				R/W-5h			

Table 7-11. FBREG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-8	FSM FB	R/W	5h	FSM FB: FSM Fallback feature Used to switch off RTL clock gating for all FSM logics used for saving power. “1010” = Disable RTL clock gating in RTL. “0101” = Enable RTL clock gating in RTL. others = no effect will be in same state as before
7-4	RESERVED	R	0h	Reserved
3-0	VBUSP FB	R/W	5h	VBUSP FB: VBUSP Fallback feature Used to switch off RTL clock gating for all VBUSP logics used for saving power. “1010” = Disable RTL clock gating in RTL. “0101” = Enable RTL clock gating in RTL. others = no effect will be in same state as before

7.2.1.4 DMASTAT Register (Offset = Ch) [reset = 0h]

DMASTAT is shown in Figure 7-21 and described in Table 7-12.

Return to Summary Table.

Channel active information

Figure 7-21. DMASTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STCH																															
R/W-0h																															

Table 7-12. DMASTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STCH	R/W	0h	0STCH: Status of DMA channels 1 channel active (means channel is currently in DMA's execution queue). 0 channel inactive. Since the DMA has two Ports only two channels can be active at a time. Writing has no effect. Note: The status of a channel currently in DMA's execution queue remains active even if emulation mode is entered or DMA is disabled via DMA_EN bit

7.2.1.5 HWCHENAS Register (Offset = 14h) [reset = 0h]

HWCHENAS is shown in Figure 7-22 and described in Table 7-13.

Return to Summary Table.

Enables a channel for HW trigger

Figure 7-22. HWCHENAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWCHENA																															
R/W-0h																															

Table 7-13. HWCHENAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWCHENA	R/W	0h	HWCHENA: HW Channel Enable Bit Writing a one to a bit enables the according channel for HW triggering. Writing a zero has no effect. An active HW DMA request can not initiate DMA transfer unless the associated HW enable bit is set. The associated HW enable bit is cleared automatically for the following conditions: 1)At the end of a block transfer if autoinitiation bit AIM in Section 1.6.3.4, "CHANNEL CONTROL Register (CHCTRL)" is not active. 2)If an AHB bus error is detected for an active channel as indicated by HRESP signal. Reading from HWCHENAS gives the status (enabled/disabled) of channels 0 through 31. 1 the associated channel is enabled for HW triggering. 0 the associated channel is disabled for HW triggering.

7.2.1.6 HWCHENAR Register (Offset = 1Ch) [reset = 0h]

HWCHENAR is shown in Figure 7-23 and described in Table 7-14.

Return to Summary Table.

Disables a channel for HW trigger

Figure 7-23. HWCHENAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWCHDIS																															
R/W-0h																															

Table 7-14. HWCHENAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWCHDIS	R/W	0h	HW Channel Disable Bit. Writing a one to a bit disables the according channel for HW triggering. Writing a zero has no effect. Reading from CHENAR gives the status (enabled/disabled) of channels 0 through 31. 1 the associated channel is enabled for HW triggering. 0 the associated channel is disabled for HW triggering. Note: All Set and Reset registers throughout DMA controller such as HWCHENAS and HWCHENAR are two logical locations mapped to one physical register.

7.2.1.7 SWCHENAS Register (Offset = 24h) [reset = 0h]

SWCHENAS is shown in Figure 7-24 and described in Table 7-15.

Return to Summary Table.

Activate a SW request

Figure 7-24. SWCHENAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWCHENA																															
R/W-0h																															

Table 7-15. SWCHENAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SWCHENA	R/W	0h	SWCHENA: SW Channel Enable Bit. Writing a one to a bit trigger a SW request on the associated channel to start DMA transaction. The associated bit is automatically cleared by the following conditions. 1)The associated bit is cleared after one frame transfer if the TTYPE bit in Section 1.6.3.4, "CHANNEL CONTROL Register (CHCTRL)" is programmed for frame transfer. 2)The associated bit is cleared after one block transfer if the corresponding TTYPE bit is programmed for block transfer and the autoinitiation bit is not enabled. 3)The control packet is modified after the pending bit is set. 4)The associated bit is cleared after one block transfer when TTYPE bit is programmed for blocks transfer and if the corresponding bit in HW Channel Enable Register (HWCHENAS) is enabled. When a channel is enabled for both HW and SW, the state machine will initiate transfers based on the SW first. After one block transfer is complete, the associated bit in the SWCHENA register is then cleared. The same channel is serviced again by a HW DMA request. 5)The associated bit is cleared if an AHB bus error is detected for an active channel as indicated by HRESP signal. Reading from SWCHENAS gives the status (enabled/disabled) of channels 0 through 31 1 the associated channel was triggered by SW request. 0 the associated channel was not triggered by SW request

7.2.1.8 SWCHENAR Register (Offset = 2Ch) [reset = 0h]

SWCHENAR is shown in Figure 7-25 and described in Table 7-16.

Return to Summary Table.

Disables a SW request

Figure 7-25. SWCHENAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWCHDIS																															
R/W-0h																															

Table 7-16. SWCHENAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SWCHDIS	R/W	0h	SWCHDIS: SW Channel Disable Bit Writing a one to a bit disables the according channel from SW trigger. Writing a zero has no effect. Reading from SWCHENAR gives the status (enabled/disabled) of channels 0 through 31. 1 the associated channel was triggered by SW. 0 the associated channel was not triggered by SW.

7.2.1.9 CHPRIOS Register (Offset = 34h) [reset = 0h]

CHPRIOS is shown in Figure 7-26 and described in Table 7-17.

Return to Summary Table.

Enables a channel

Figure 7-26. CHPRIOS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPH																															
R/W-0h																															

Table 7-17. CHPRIOS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPH	R/W	0h	CPH: Channel Priority Set Bit. Writing a one to a bit assigns the according channel to the high priority queue. Writing a zero has no effect. Reading from CHPRIOS gives the status (high priority 1 / low priority 0) of channels 0 through 31. 1 the associated channel is assigned to high priority queue. 0 the associated channel is assigned to low priority queue.

7.2.1.10 CHPRIOR Register (Offset = 3Ch) [reset = 0h]

CHPRIOR is shown in Figure 7-27 and described in Table 7-18.

Return to Summary Table.

Disables a channel

Figure 7-27. CHPRIOR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPL																															
R/W-0h																															

Table 7-18. CHPRIOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPL	R/W	0h	CPL: Channel Priority Reset Bit Writing a one to a bit assigns the according channel to the low priority queue. Writing a zero has no effect. Reading from CHPRIOR gives the status (high priority 1 / low priority 0) of channels 0 through 31. 1 the associated channel is assigned to high priority queue. 0 the associated channel is assigned to low priority queue.

7.2.1.11 GCHIENAS Register (Offset = 44h) [reset = 0h]

GCHIENAS is shown in Figure 7-28 and described in Table 7-19.

Return to Summary Table.

Enables the specific channel interrupts

Figure 7-28. GCHIENAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GCHIE																															
R/W-0h																															

Table 7-19. GCHIENAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GCHIE	R/W	0h	GCHIE: Global Channel Interrupt Enable Bit. Writing a one to a bit enables the according channel interrupt. Writing a zero has no effect. Reading from GCHIENAS gives the status (enabled/disabled) of channels 0 through 31. 1 the associated channel is enabled for interrupt. 0 the associated channel is disabled for interrupt.

7.2.1.12 GCHIENAR Register (Offset = 4Ch) [reset = 0h]

GCHIENAR is shown in Figure 7-29 and described in Table 7-20.

Return to Summary Table.

Disables the specific channel interrupts

Figure 7-29. GCHIENAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GCHID																															
R/W-0h																															

Table 7-20. GCHIENAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GCHID	R/W	0h	GCHID: Global Channel Interrupt Disable Bit Writing a one to a bit disables the according channel interrupt. Writing a zero has no effect. Reading from GCHIENAR gives the status (enabled/disabled) of channels 0 through 31. 1 the associated channel is enabled for interrupt. 0 the associated channel is disabled for interrupt

7.2.1.13 DREQASI0 Register (Offset = 54h) [reset = 00010203h]

DREQASI0 is shown in Figure 7-30 and described in Table 7-21.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-30. DREQASI0 Register

31	30	29	28	27	26	25	24
RESERVED		CH0ASI_5_0					
R-0h		R/W-0h					
23	22	21	20	19	18	17	16
RESERVED		CH1ASI_5_0					
R-0h		R/W-1h					
15	14	13	12	11	10	9	8
RESERVED		CH2ASI_5_0					
R-0h		R/W-2h					
7	6	5	4	3	2	1	0
RESERVED		CH3ASI_5_0					
R-0h		R/W-3h					

Table 7-21. DREQASI0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH0ASI_5_0	R/W	0h	CH0ASI: Channel 0 Assignment Bits This bit field chooses the DMA request assignment for channel 0. Writing 0 means DMA request line 0 coming in to the DMA will trigger channel 0. Writing 3F means DMA request line 63 will trigger channel 0.
23-22	RESERVED	R	0h	Reserved
21-16	CH1ASI_5_0	R/W	1h	CH1ASI: Channel 1 Assignment Bits This bit field chooses the DMA request assignment for channel 1.
15-14	RESERVED	R	0h	Reserved
13-8	CH2ASI_5_0	R/W	2h	CH2ASI: Channel 2 Assignment Bits This bit field chooses the DMA request assignment for channel 2.
7-6	RESERVED	R	0h	Reserved
5-0	CH3ASI_5_0	R/W	3h	CH3ASI: Channel 3 Assignment Bits. This bit field chooses the DMA request assignment for channel 3

7.2.1.14 DREQASI1 Register (Offset = 58h) [reset = 04050607h]

DREQASI1 is shown in Figure 7-31 and described in Table 7-22.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-31. DREQASI1 Register

31	30	29	28	27	26	25	24
RESERVED				CH4ASI_5_0			
R-0h				R/W-4h			
23	22	21	20	19	18	17	16
RESERVED				CH5ASI_5_0			
R-0h				R/W-5h			
15	14	13	12	11	10	9	8
RESERVED				CH6ASI_5_0			
R-0h				R/W-6h			
7	6	5	4	3	2	1	0
RESERVED				CH7ASI_5_0			
R-0h				R/W-7h			

Table 7-22. DREQASI1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH4ASI_5_0	R/W	4h	CH4ASI: Channel 4 Assignment Bits This bit field chooses the DMA request assignment for channel 4. .
23-22	RESERVED	R	0h	Reserved
21-16	CH5ASI_5_0	R/W	5h	CH5ASI: Channel 5 Assignment Bits This bit field chooses the DMA request assignment for channel 5
15-14	RESERVED	R	0h	Reserved
13-8	CH6ASI_5_0	R/W	6h	CH6ASI: Channel 6 Assignment Bits This bit field chooses the DMA request assignment for channel 6
7-6	RESERVED	R	0h	Reserved
5-0	CH7ASI_5_0	R/W	7h	CH7ASI: Channel 7 Assignment Bits. This bit field chooses the DMA request assignment for channel 7

7.2.1.15 DREQASI2 Register (Offset = 5Ch) [reset = 08090A0Bh]

DREQASI2 is shown in Figure 7-32 and described in Table 7-23.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-32. DREQASI2 Register

31	30	29	28	27	26	25	24
RESERVED		CH8ASI_5_0					
R-0h		R/W-8h					
23	22	21	20	19	18	17	16
RESERVED		CH9ASI_5_0					
R-0h		R/W-9h					
15	14	13	12	11	10	9	8
RESERVED		CH10ASI_5_0					
R-0h		R/W-Ah					
7	6	5	4	3	2	1	0
RESERVED		CH11ASI_5_0					
R-0h		R/W-Bh					

Table 7-23. DREQASI2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH8ASI_5_0	R/W	8h	CH8ASI: Channel 8 Assignment Bits This bit field chooses the DMA request assignment for channel 8
23-22	RESERVED	R	0h	Reserved
21-16	CH9ASI_5_0	R/W	9h	CH9ASI: Channel 9 Assignment Bits This bit field chooses the DMA request assignment for channel 9
15-14	RESERVED	R	0h	Reserved
13-8	CH10ASI_5_0	R/W	Ah	CH10ASI: Channel 10 Assignment Bits This bit field chooses the DMA request assignment for channel 10
7-6	RESERVED	R	0h	Reserved
5-0	CH11ASI_5_0	R/W	Bh	CH11ASI: Channel 11 Assignment Bits. This bit field chooses the DMA request assignment for channel 11

7.2.1.16 DREQASI3 Register (Offset = 60h) [reset = 0C0D0E0Fh]

DREQASI3 is shown in Figure 7-33 and described in Table 7-24.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-33. DREQASI3 Register

31	30	29	28	27	26	25	24
RESERVED		CH12ASI_5_0					
R-0h		R/W-Ch					
23	22	21	20	19	18	17	16
RESERVED		CH13ASI_5_0					
R-0h		R/W-Dh					
15	14	13	12	11	10	9	8
RESERVED		CH14ASI_5_0					
R-0h		R/W-Eh					
7	6	5	4	3	2	1	0
RESERVED		CH15ASI_5_0					
R-0h		R/W-Fh					

Table 7-24. DREQASI3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH12ASI_5_0	R/W	Ch	CH12ASI: Channel 12 Assignment Bits This bit field chooses the DMA request assignment for channel 12
23-22	RESERVED	R	0h	Reserved
21-16	CH13ASI_5_0	R/W	Dh	CH13ASI: Channel 13 Assignment Bits This bit field chooses the DMA request assignment for channel 13
15-14	RESERVED	R	0h	Reserved
13-8	CH14ASI_5_0	R/W	Eh	CH14ASI: Channel 14 Assignment Bits This bit field chooses the DMA request assignment for channel 14
7-6	RESERVED	R	0h	Reserved
5-0	CH15ASI_5_0	R/W	Fh	CH15ASI: Channel 15 Assignment Bits. This bit field chooses the DMA request assignment for channel 15

7.2.1.17 DREQASI4 Register (Offset = 64h) [reset = 10111213h]

DREQASI4 is shown in Figure 7-34 and described in Table 7-25.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-34. DREQASI4 Register

31	30	29	28	27	26	25	24
RESERVED		CH16ASI_5_0					
R-0h		R/W-10h					
23	22	21	20	19	18	17	16
RESERVED		CH17ASI_5_0					
R-0h		R/W-11h					
15	14	13	12	11	10	9	8
RESERVED		CH18ASI_5_0					
R-0h		R/W-12h					
7	6	5	4	3	2	1	0
RESERVED		CH19ASI_5_0					
R-0h		R/W-13h					

Table 7-25. DREQASI4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH16ASI_5_0	R/W	10h	CH16ASI: Channel 16 Assignment Bits This bit field chooses the DMA request assignment for channel 16
23-22	RESERVED	R	0h	Reserved
21-16	CH17ASI_5_0	R/W	11h	CH17ASI: Channel 17 Assignment Bits This bit field chooses the DMA request assignment for channel 17
15-14	RESERVED	R	0h	Reserved
13-8	CH18ASI_5_0	R/W	12h	CH18ASI: Channel 18 Assignment Bits This bit field chooses the DMA request assignment for channel 18
7-6	RESERVED	R	0h	Reserved
5-0	CH19ASI_5_0	R/W	13h	CH18ASI: Channel 19 Assignment Bits. This bit field chooses the DMA request assignment for channel 19

7.2.1.18 DREQASI5 Register (Offset = 68h) [reset = 14151617h]

DREQASI5 is shown in Figure 7-35 and described in Table 7-26.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-35. DREQASI5 Register

31	30	29	28	27	26	25	24
RESERVED		CH20ASI_5_0					
R-0h		R/W-14h					
23	22	21	20	19	18	17	16
RESERVED		CH21ASI_5_0					
R-0h		R/W-15h					
15	14	13	12	11	10	9	8
RESERVED		CH22ASI_5_0					
R-0h		R/W-16h					
7	6	5	4	3	2	1	0
RESERVED		CH23ASI_5_0					
R-0h		R/W-17h					

Table 7-26. DREQASI5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH20ASI_5_0	R/W	14h	CH20ASI: Channel 20 Assignment Bits This bit field chooses the DMA request assignment for channel 20
23-22	RESERVED	R	0h	Reserved
21-16	CH21ASI_5_0	R/W	15h	CH21ASI: Channel 21 Assignment Bits This bit field chooses the DMA request assignment for channel 21
15-14	RESERVED	R	0h	Reserved
13-8	CH22ASI_5_0	R/W	16h	CH22ASI: Channel 22 Assignment Bits This bit field chooses the DMA request assignment for channel 22
7-6	RESERVED	R	0h	Reserved
5-0	CH23ASI_5_0	R/W	17h	CH23ASI: Channel 23 Assignment Bits. This bit field chooses the DMA request assignment for channel 23

7.2.1.19 DREQASI6 Register (Offset = 6Ch) [reset = 18191A1Bh]

DREQASI6 is shown in Figure 7-36 and described in Table 7-27.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-36. DREQASI6 Register

31	30	29	28	27	26	25	24
RESERVED		CH24ASI_5_0					
R-0h		R/W-18h					
23	22	21	20	19	18	17	16
RESERVED		CH25ASI_5_0					
R-0h		R/W-19h					
15	14	13	12	11	10	9	8
RESERVED		CH26ASI_5_0					
R-0h		R/W-1Ah					
7	6	5	4	3	2	1	0
RESERVED		CH27ASI_5_0					
R-0h		R/W-1Bh					

Table 7-27. DREQASI6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH24ASI_5_0	R/W	18h	CH24ASI: Channel 24 Assignment Bits This bit field chooses the DMA request assignment for channel 24
23-22	RESERVED	R	0h	Reserved
21-16	CH25ASI_5_0	R/W	19h	CH25SI: Channel 25 Assignment Bits This bit field chooses the DMA request assignment for channel 25
15-14	RESERVED	R	0h	Reserved
13-8	CH26ASI_5_0	R/W	1Ah	CH26ASI: Channel 26 Assignment Bits This bit field chooses the DMA request assignment for channel 26
7-6	RESERVED	R	0h	Reserved
5-0	CH27ASI_5_0	R/W	1Bh	CH27ASI: Channel 27 Assignment Bits. This bit field chooses the DMA request assignment for channel 27

7.2.1.20 DREQASI7 Register (Offset = 70h) [reset = 1C1D1E1Fh]

DREQASI7 is shown in Figure 7-37 and described in Table 7-28.

Return to Summary Table.

DMA channel to request line assignment

Figure 7-37. DREQASI7 Register

31	30	29	28	27	26	25	24
RESERVED		CH28ASI_5_0					
R-0h		R/W-1Ch					
23	22	21	20	19	18	17	16
RESERVED		CH29ASI_5_0					
R-0h		R/W-1Dh					
15	14	13	12	11	10	9	8
RESERVED		CH30ASI_5_0					
R-0h		R/W-1Eh					
7	6	5	4	3	2	1	0
RESERVED		CH31ASI_5_0					
R-0h		R/W-1Fh					

Table 7-28. DREQASI7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29-24	CH28ASI_5_0	R/W	1Ch	CH28ASI: Channel 28 Assignment Bits This bit field chooses the DMA request assignment for channel 28
23-22	RESERVED	R	0h	Reserved
21-16	CH29ASI_5_0	R/W	1Dh	CH29SI: Channel 29 Assignment Bits This bit field chooses the DMA request assignment for channel 29
15-14	RESERVED	R	0h	Reserved
13-8	CH30ASI_5_0	R/W	1Eh	CH30ASI: Channel 30 Assignment Bits This bit field chooses the DMA request assignment for channel 30
7-6	RESERVED	R	0h	Reserved
5-0	CH31ASI_5_0	R/W	1Fh	CH31ASI: Channel 31 Assignment Bits. This bit field chooses the DMA request assignment for channel 31

7.2.1.21 PAR0 Register (Offset = 94h) [reset = 0h]

PAR0 is shown in Figure 7-38 and described in Table 7-29.

Return to Summary Table.

DMA channel to port assignment

Figure 7-38. PAR0 Register

31	30	29	28	27	26	25	24
RESERVED	CH0PA_2_0			RESERVED	CH1PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	CH2PA_2_0			RESERVED	CH3PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	CH4PA_2_0			RESERVED	CH5PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	CH6PA_2_0			RESERVED	CH7PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 7-29. PAR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	CH0PA_2_0	R/W	0h	CH0PA These bit fields determine which port channel 0 is assigned. 1xx Port B 011 Port A2 only (Not valid for 16xx) 010 Port A1 only (Not valid for 16xx) 001 Port A1/2 combined, A2 read / A1 write (Not valid for 16xx) 000 Port A1/2 combined, A1 read / A2 write (Not valid for 16xx)
27	RESERVED	R	0h	Reserved
26-24	CH1PA_2_0	R/W	0h	CH1PA These bit fields determine which port channel 1 is assigned
23	RESERVED	R	0h	Reserved
22-20	CH2PA_2_0	R/W	0h	CH2PA These bit fields determine which port channel 2 is assigned
19	RESERVED	R	0h	Reserved
18-16	CH3PA_2_0	R/W	0h	CH3PA These bit fields determine which port channel 3 is assigned
15	RESERVED	R	0h	Reserved
14-12	CH4PA_2_0	R/W	0h	CH4PA These bit fields determine which port channel 4 is assigned
11	RESERVED	R	0h	Reserved
10-8	CH5PA_2_0	R/W	0h	CH5PA These bit fields determine which port channel 5 is assigned
7	RESERVED	R	0h	Reserved
6-4	CH6PA_2_0	R/W	0h	CH6PA These bit fields determine which port channel 6 is assigned
3	RESERVED	R	0h	Reserved
2-0	CH7PA_2_0	R/W	0h	CH7PA These bit fields determine which port channel 7 is assigned

7.2.1.22 PAR1 Register (Offset = 98h) [reset = 0h]

PAR1 is shown in Figure 7-39 and described in Table 7-30.

Return to Summary Table.

DMA channel to port assignment

Figure 7-39. PAR1 Register

31	30	29	28	27	26	25	24
RESERVED	CH8PA_2_0			RESERVED	CH9PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	CH10PA_2_0			RESERVED	CH11PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	CH12PA_2_0			RESERVED	CH13PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	CH14PA_2_0			RESERVED	CH15PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 7-30. PAR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	CH8PA_2_0	R/W	0h	CH8PA These bit fields determine which port channel 8 is assigned. 1xx Port B 011 Port A2 only (Not valid for 16xx) 010 Port A1 only (Not valid for 16xx) 001 Port A1/2 combined, A2 read / A1 write (Not valid for 16xx) 000 Port A1/2 combined, A1 read / A2 write (Not valid for 16xx)
27	RESERVED	R	0h	Reserved
26-24	CH9PA_2_0	R/W	0h	CH9PA These bit fields determine which port channel 9 is assigned
23	RESERVED	R	0h	Reserved
22-20	CH10PA_2_0	R/W	0h	CH10PA These bit fields determine which port channel 10 is assigned
19	RESERVED	R	0h	Reserved
18-16	CH11PA_2_0	R/W	0h	CH11PA These bit fields determine which port channel 11 is assigned
15	RESERVED	R	0h	Reserved
14-12	CH12PA_2_0	R/W	0h	CH12PA These bit fields determine which port channel 12 is assigned
11	RESERVED	R	0h	Reserved
10-8	CH13PA_2_0	R/W	0h	CH13PA These bit fields determine which port channel 13 is assigned
7	RESERVED	R	0h	Reserved
6-4	CH14PA_2_0	R/W	0h	CH14PA These bit fields determine which port channel 14 is assigned
3	RESERVED	R	0h	Reserved
2-0	CH15PA_2_0	R/W	0h	CH15PA These bit fields determine which port channel 15 is assigned

7.2.1.23 PAR2 Register (Offset = 9Ch) [reset = 0h]

PAR2 is shown in Figure 7-40 and described in Table 7-31.

Return to Summary Table.

DMA channel to port assignment

Figure 7-40. PAR2 Register

31	30	29	28	27	26	25	24
RESERVED	CH16PA_2_0			RESERVED	CH17PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	CH18PA_2_0			RESERVED	CH19PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	CH20PA_2_0			RESERVED	CH21PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	CH22PA_2_0			RESERVED	CH23PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 7-31. PAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	CH16PA_2_0	R/W	0h	CH16PA These bit fields determine which port channel 16 is assigned. 1xx Port B 011 Port A2 only (Not valid for 16xx) 010 Port A1 only (Not valid for 16xx) 001 Port A1/2 combined, A2 read / A1 write (Not valid for 16xx) 000 Port A1/2 combined, A1 read / A2 write (Not valid for 16xx)
27	RESERVED	R	0h	Reserved
26-24	CH17PA_2_0	R/W	0h	CH17PA These bit fields determine which port channel 17 is assigned
23	RESERVED	R	0h	Reserved
22-20	CH18PA_2_0	R/W	0h	CH18PA These bit fields determine which port channel 18 is assigned
19	RESERVED	R	0h	Reserved
18-16	CH19PA_2_0	R/W	0h	CH19PA These bit fields determine which port channel 19 is assigned
15	RESERVED	R	0h	Reserved
14-12	CH20PA_2_0	R/W	0h	CH20PA These bit fields determine which port channel 20 is assigned
11	RESERVED	R	0h	Reserved
10-8	CH21PA_2_0	R/W	0h	CH21PA These bit fields determine which port channel 21 is assigned
7	RESERVED	R	0h	Reserved
6-4	CH22PA_2_0	R/W	0h	CH22PA These bit fields determine which port channel 22 is assigned
3	RESERVED	R	0h	Reserved
2-0	CH23PA_2_0	R/W	0h	CH23PA These bit fields determine which port channel 23 is assigned

7.2.1.24 PAR3 Register (Offset = A0h) [reset = 0h]

PAR3 is shown in Figure 7-41 and described in Table 7-32.

Return to Summary Table.

DMA channel to port assignment

Figure 7-41. PAR3 Register

31	30	29	28	27	26	25	24
RESERVED	CH24PA_2_0			RESERVED	CH25PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	CH26PA_2_0			RESERVED	CH27PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	CH28PA_2_0			RESERVED	CH29PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	CH30PA_2_0			RESERVED	CH31PA_2_0		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 7-32. PAR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30-28	CH24PA_2_0	R/W	0h	CH24PA These bit fields determine which port channel 24 is assigned. 1xx Port B 011 Port A2 only (Not valid for 16xx) 010 Port A1 only (Not valid for 16xx) 001 Port A1/2 combined, A2 read / A1 write (Not valid for 16xx) 000 Port A1/2 combined, A1 read / A2 write (Not valid for 16xx)
27	RESERVED	R	0h	Reserved
26-24	CH25PA_2_0	R/W	0h	CH25PA These bit fields determine which port channel 25 is assigned
23	RESERVED	R	0h	Reserved
22-20	CH26PA_2_0	R/W	0h	CH26PA These bit fields determine which port channel 26 is assigned
19	RESERVED	R	0h	Reserved
18-16	CH27PA_2_0	R/W	0h	CH27PA These bit fields determine which port channel 27 is assigned
15	RESERVED	R	0h	Reserved
14-12	CH28PA_2_0	R/W	0h	CH28PA These bit fields determine which port channel 28 is assigned
11	RESERVED	R	0h	Reserved
10-8	CH29PA_2_0	R/W	0h	CH29PA These bit fields determine which port channel 28 is assigned
7	RESERVED	R	0h	Reserved
6-4	CH30PA_2_0	R/W	0h	CH30PA These bit fields determine which port channel 30 is assigned
3	RESERVED	R	0h	Reserved
2-0	CH31PA_2_0	R/W	0h	CH31PA These bit fields determine which port channel 31 is assigned

7.2.1.25 FTCMAP Register (Offset = B4h) [reset = 0h]

FTCMAP is shown in Figure 7-42 and described in Table 7-33.

Return to Summary Table.

Assignment of interrupt lines to either ARM CPU or DSP CPU

Figure 7-42. FTCMAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTCAB																															
R/W-0h																															

Table 7-33. FTCMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FTCAB	R/W	0h	FTCAB: Frame Transfer Complete Interrupt to Group A or Group B 1 routes FTC interrupt of the corresponding channel to Group B. 0 routes FTC interrupt of the corresponding channel to Group A. Note: Group A interrupts (FTC, LFS, HBC, BTC and BER) are routed to ARM CPU. Group B interrupts (FTC, LFS, HBC, BTC and BER) are routed to DSP CPU.

7.2.1.26 LFSMAP Register (Offset = BCh) [reset = 0h]

LFSMAP is shown in Figure 7-43 and described in Table 7-34.

Return to Summary Table.

Assignment of interrupt lines to either ARM CPU or DSP CPU

Figure 7-43. LFSMAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFSAB																															
R/W-0h																															

Table 7-34. LFSMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LFSAB	R/W	0h	LFSAB: Last Frame Started Interrupt to Group A or Group B 1 routes LFS interrupt of the corresponding channel to Group B. 0 routes LFS interrupt of the corresponding channel to Group A.

7.2.1.27 HBCMAP Register (Offset = C4h) [reset = 0h]

HBCMAP is shown in Figure 7-44 and described in Table 7-35.

Return to Summary Table.

Assignment of interrupt lines to either ARM CPU or DSP CPU

Figure 7-44. HBCMAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBCAB																															
R/W-0h																															

Table 7-35. HBCMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HBCAB	R/W	0h	HBCAB: Half Block Complete Interrupt to Group A or Group B 1 routes HBC interrupt of the corresponding channel to Group B. 0 routes HBC interrupt of the corresponding channel to Group A.

7.2.1.28 BTCMAP Register (Offset = CCh) [reset = 0h]

BTCMAP is shown in Figure 7-45 and described in Table 7-36.

Return to Summary Table.

Assignment of interrupt lines to either ARM CPU or DSP CPU

Figure 7-45. BTCMAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTCAB																															
R/W-0h																															

Table 7-36. BTCMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BTCAB	R/W	0h	BTCAB: Block Transfer Complete Interrupt to Group A or Group B 1 routes BTC interrupt of the corresponding channel to Group B. 0 routes BTC interrupt of the corresponding channel to Group A.

7.2.1.29 BERMAP Register (Offset = D4h) [reset = 0h]

BERMAP is shown in Figure 7-46 and described in Table 7-37.

Return to Summary Table.

Assignment of interrupt lines to either ARM CPU or DSP CPU

Figure 7-46. BERMAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BERAB																															
R/W-0h																															

Table 7-37. BERMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BERAB	R/W	0h	BERAB: Bus Error Interrupt to Group A or Group B. 1 routes BER interrupt of the corresponding channel to Group B. 0 routes BER interrupt of the corresponding channel to Group A.

7.2.1.30 FTCINTENAS Register (Offset = DCh) [reset = 0h]

FTCINTENAS is shown in Figure 7-47 and described in Table 7-38.

Return to Summary Table.

Enables a FTC interrupt for channels 0-31

Figure 7-47. FTCINTENAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTCINTENA																															
R/W-0h																															

Table 7-38. FTCINTENAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FTCINTENA	R/W	0h	FTCINTENA: FTC (Frame Transfer Complete) Interrupt Enable Bit. Writing a one to a bit enables the interrupt of the according channel. Writing a zero has no effect. Reading from FTCINTENAS gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated FTC interrupt of a channel is enabled. 0 the associated FTC interrupt of a channel is disabled.

7.2.1.31 FTCINTENAR Register (Offset = E4h) [reset = 0h]

FTCINTENAR is shown in Figure 7-48 and described in Table 7-39.

Return to Summary Table.

Disables a FTC interrupt for channels 0-31

Figure 7-48. FTCINTENAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTCINTDIS																															
R/W-0h																															

Table 7-39. FTCINTENAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FTCINTDIS	R/W	0h	FTCINTDIS: FTC (Frame Transfer Complete) Interrupt Disable Bit. Writing a one to a bit disables the interrupt of the according channel. Writing a zero has no effect. Reading from FTCINTENAR gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated FTC interrupt of a channel is enabled. 0 the associated FTC interrupt of a channel is disabled.

7.2.1.32 LFSINTENAS Register (Offset = ECh) [reset = 0h]

LFSINTENAS is shown in Figure 7-49 and described in Table 7-40.

Return to Summary Table.

Enables a LFS interrupt for channels 0-31

Figure 7-49. LFSINTENAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFSINTENA																															
R/W-0h																															

Table 7-40. LFSINTENAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LFSINTENA	R/W	0h	LFSINTENA: LFS (last Frame Started) Interrupt Enable Bit. Writing a one to a bit enables the interrupt of the according channel. Writing a zero has no effect Reading from LFSINTENAS gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated LFS interrupt of a channel is enabled. 0 the associated LFS interrupt of a channel is disabled

7.2.1.33 LFSINTENAR Register (Offset = F4h) [reset = 0h]

LFSINTENAR is shown in Figure 7-50 and described in Table 7-41.

Return to Summary Table.

Disables a LFS interrupt for channels 0-31

Figure 7-50. LFSINTENAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFSINTDIS																															
R/W-0h																															

Table 7-41. LFSINTENAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LFSINTDIS	R/W	0h	LFSINTDIS: LFS (Last Frame Started) Interrupt Disable Bit. Writing a one to a bit disables the interrupt of the according channel. Writing a zero has no effect. Reading from LFSINTENAR gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated LFS interrupt of a channel is enabled. 0 the associated LFS interrupt of a channel is disabled

7.2.1.34 HBCINTENAS Register (Offset = FCh) [reset = 0h]

HBCINTENAS is shown in Figure 7-51 and described in Table 7-42.

Return to Summary Table.

Enables a HBC interrupt for channels 0-31

Figure 7-51. HBCINTENAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBCINTENA																															
R/W-0h																															

Table 7-42. HBCINTENAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HBCINTENA	R/W	0h	HBCINTENA: HBC (Half Block Complete) Interrupt Enable Bit. Writing a one to a bit enables the interrupt of the according channel. Writing a zero has no effect. Reading from HBCINTENAS gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated HBC interrupt of a channel is enabled. 0 the associated HBC interrupt of a channel is disabled

7.2.1.35 HBCINTENAR Register (Offset = 104h) [reset = 0h]

HBCINTENAR is shown in Figure 7-52 and described in Table 7-43.

Return to Summary Table.

Disables a HBC interrupt for channels 0-31

Figure 7-52. HBCINTENAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBCINTDIS																															
R/W-0h																															

Table 7-43. HBCINTENAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HBCINTDIS	R/W	0h	HBCINTDIS: HBC (Half Block Complete) Interrupt Disable Bit. Writing a one to a bit disables the interrupt of the according channel. Writing a zero has no effect Reading from HBCINTENAR gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated HBC interrupt of a channel is enabled. 0 the associated HBC interrupt of a channel is disabled

7.2.1.36 BTCINTENAS Register (Offset = 10Ch) [reset = 0h]

BTCINTENAS is shown in Figure 7-53 and described in Table 7-44.

Return to Summary Table.

Enables a BTC interrupt for channels 0-31

Figure 7-53. BTCINTENAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTCINTENA																															
R/W-0h																															

Table 7-44. BTCINTENAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BTCINTENA	R/W	0h	BTCINTENA: BTC (Block Transfer Complete) Interrupt Enable Bit. Writing a one to a bit enables the interrupt of the according channel. Writing a zero has no effect. Reading from BTCINTENAS gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated BTC interrupt of a channel is enabled. 0 the associated BTC interrupt of a channel is disabled.

7.2.1.37 BTCINTENAR Register (Offset = 114h) [reset = 0h]

BTCINTENAR is shown in Figure 7-54 and described in Table 7-45.

Return to Summary Table.

Disables a BTC interrupt for channels 0-31

Figure 7-54. BTCINTENAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTCINTDIS																															
R/W-0h																															

Table 7-45. BTCINTENAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BTCINTDIS	R/W	0h	BTCINTDIS: BTC (Block Transfer Complete) Interrupt Disable Bit. Writing a one to a bit disables the interrupt of the according channel. Writing a zero has no effect. Reading from BTCINTENAR gives the status (interrupt enabled/disabled) of channels 0 through 31. 1 the associated BTC interrupt of a channel is enabled. 0 the associated BTC interrupt of a channel is disabled

7.2.1.38 GINTFLAG Register (Offset = 11Ch) [reset = 0h]

GINTFLAG is shown in Figure 7-55 and described in Table 7-46.

Return to Summary Table.

Each flag bit is an OR of five different interrupt types of the same channel

Figure 7-55. GINTFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GINT																															
R/W-0h																															

Table 7-46. GINTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GINT	R/W	0h	GINT: Global Interrupt Flags. Global interrupt flag bit is a OR function of FTC, LFS, HBC, BTC and BER interrupt flags. When a bit is active, it indicates that one or more of the five interrupt types is pending of the corresponding channel

7.2.1.39 FTCFLAG Register (Offset = 124h) [reset = 0h]

FTCFLAG is shown in Figure 7-56 and described in Table 7-47.

Return to Summary Table.

FTC Interrupt flags Channels 0-31

Figure 7-56. FTCFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTC																															
R/W-0h																															

Table 7-47. FTCFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FTC	R/W	0h	FTCI: Frame Transfer Complete Flags. If the bit is set a FTC Interrupt of the corresponding channels pending. Writing a 1 clears the according flag. Writing a zero has no effect. Reading from the respected Interrupt Channel Offset register also clears the corresponding flag. 1 the associated FTC interrupt of a channel is pending. 0 the associated FTC interrupt of a channel is not pending. The state of the flag bit can be polled even if the corresponding interrupt enable bit is cleared

7.2.1.40 LFSFLAG Register (Offset = 12Ch) [reset = 0h]

LFSFLAG is shown in Figure 7-57 and described in Table 7-48.

Return to Summary Table.

LFS Interrupt flags Channels 0-31

Figure 7-57. LFSFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFSI																															
R/W-0h																															

Table 7-48. LFSFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LFSI	R/W	0h	LFSI: Last Frame Transfer Started Flags If the bit is set a LFS Interrupt of the corresponding channels pending. Writing a 1 clears the according flag. Writing a zero has no effect. Reading from the respected Interrupt Channel Offset register also clears the corresponding flag. 1 the associated LFS interrupt of a channel is pending. 0 the associated LFS interrupt of a channel is not pending. The state of the flag bit can be polled even if the corresponding interrupt enable bit is cleared

7.2.1.41 HBCFLAG Register (Offset = 134h) [reset = 0h]

HBCFLAG is shown in Figure 7-58 and described in Table 7-49.

Return to Summary Table.

HBC Interrupt flags Channels 0-31

Figure 7-58. HBCFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBCI																															
R/W-0h																															

Table 7-49. HBCFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HBCI	R/W	0h	HBCI: Half of Block Transfer Complete Flags. If the bit is set a HBC Interrupt of the corresponding channels pending. Writing a 1 clears the according flag. Writing a zero has no effect. Reading from the respected Interrupt Channel Offset register also clears the corresponding flag. 1 the associated HBC interrupt of a channel is pending. 0 the associated HBC interrupt of a channel is not pending. The state of the flag bit can be polled even if the corresponding interrupt enable bit is cleared

7.2.1.42 BTCFLAG Register (Offset = 13Ch) [reset = 0h]

BTCFLAG is shown in Figure 7-59 and described in Table 7-50.

Return to Summary Table.

BTC Interrupt flags Channels 0-31

Figure 7-59. BTCFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTCI																															
R/W-0h																															

Table 7-50. BTCFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BTCI	R/W	0h	BTCI: Block Transfer Complete Flags. If the bit is set a BTC Interrupt of the corresponding channels pending. Writing a 1 clears the according flag. Writing a zero has no effect. Reading from the respected Interrupt Channel Offset register also clears the corresponding flag. 1 the associated BTC interrupt of a channel is pending. 0 the associated BTC interrupt of a channel is not pending. The state of the flag bit can be polled even if the corresponding interrupt enable bit is cleared

7.2.1.43 BERFLAG Register (Offset = 144h) [reset = 0h]

BERFLAG is shown in Figure 7-60 and described in Table 7-51.

Return to Summary Table.

BER Interrupt flags Channels 0-31

Figure 7-60. BERFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BERI																															
R/W-0h																															

Table 7-51. BERFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BERI	R/W	0h	BERI: Bus Error Flags. If the bit is set a BER Interrupt of the corresponding channels pending. Writing a 1 clears the according flag. Writing a zero has no effect. Reading from the respected Interrupt Channel Offset register also clears the corresponding flag. 1 the associated BER interrupt of a channel is pending. 0 the associated BER interrupt of a channel is not pending. The state of the flag bit can be polled even if the corresponding interrupt enable bit is cleared.

7.2.1.44 FTCAOFFSET Register (Offset = 14Ch) [reset = 0h]

FTCAOFFSET is shown in Figure 7-61 and described in Table 7-52.

Return to Summary Table.

Channel causing FTC Interrupt on Group A

Figure 7-61. FTCAOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		FTCA_5_0					
R-0h								R-0h		R-0h					

Table 7-52. FTCAOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	FTCA_5_0	R	0h	FTCA: Channel causing FTC (Frame Transfer Complete) Interrupt Group A. Contains the channel number of the pending interrupt for group A if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer to Table 1–10. If FTCA is zero it means no interrupt is pending. FTCA = 1, means channel 0 is pending. FTCA = 2, means channel 1 is pending and so on... FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | Channel 31 FTCA 0x20

7.2.1.45 LFSAOFFSET Register (Offset = 150h) [reset = 0h]

LFSAOFFSET is shown in Figure 7-62 and described in Table 7-53.

Return to Summary Table.

Channel causing LFS Interrupt on Group A

Figure 7-62. LFSAOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		LFSA_5_0					
R-0h								R-0h		R-0h					

Table 7-53. LFSAOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	6sbz. These bits should be programmed as zero.
5-0	LFSA_5_0	R	0h	LFSA: Channel causing LFS (Last Frame Started) Interrupt Group A. Contains the channel number of the pending interrupt for group A if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer below table FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | | Channel 31 FTCA 0x20

7.2.1.46 HBCAOFFSET Register (Offset = 154h) [reset = 0h]

HBCAOFFSET is shown in Figure 7-63 and described in Table 7-54.

Return to Summary Table.

Channel causing HBC Interrupt on Group A

Figure 7-63. HBCAOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		HBCA_5_0					
R-0h								R-0h		R-0h					

Table 7-54. HBCAOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	HBCA_5_0	R	0h	HBCA: Channel causing HBC (Half Block Complete) Interrupt Group A. Contains the channel number of the pending interrupt for group A if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer to below table FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | Channel 31 FTCA 0x20

7.2.1.47 BTCAOFFSET Register (Offset = 158h) [reset = 0h]

BTCAOFFSET is shown in Figure 7-64 and described in Table 7-55.

Return to Summary Table.

Channel causing BTC Interrupt on Group A

Figure 7-64. BTCAOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		BTCA_5_0					
R-0h								R-0h		R-0h					

Table 7-55. BTCAOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	BTCA_5_0	R	0h	BTCA: Channel causing BTC (Block Transfer Complete) Interrupt Group A. Contains the channel number of the pending interrupt for group A if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority.

7.2.1.48 BERAOFFSET Register (Offset = 15Ch) [reset = 0h]

BERAOFFSET is shown in Figure 7-65 and described in Table 7-56.

Return to Summary Table.

Channel causing BER Interrupt on Group A

Figure 7-65. BERAOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		BERA_5_0					
R-0h								R-0h		R-0h					

Table 7-56. BERAOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	BERA_5_0	R	0h	BERA: Channel causing BER (Bus Error) Interrupt Group A. Contains the channel number of the pending interrupt for group A if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority

7.2.1.49 FTCBOFFSET Register (Offset = 160h) [reset = 0h]

FTCBOFFSET is shown in Figure 7-66 and described in Table 7-57.

Return to Summary Table.

Channel causing FTC Interrupt on Group B

Figure 7-66. FTCBOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		FTCB_5_0					
R-0h								R-0h		R-0h					

Table 7-57. FTCBOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	FTCB_5_0	R	0h	FTCB: Channel causing FTC (Frame Transfer Complete) Interrupt Group B. Contains the channel number of the pending interrupt for group B. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer to below table FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | | Channel 31 FTCA 0x20

7.2.1.50 LFSBOFFSET Register (Offset = 164h) [reset = 0h]

LFSBOFFSET is shown in Figure 7-67 and described in Table 7-58.

Return to Summary Table.

Channel causing LFS Interrupt on Group B

Figure 7-67. LFSBOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		LFSB_5_0					
R-0h								R-0h		R-0h					

Table 7-58. LFSBOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	LFSB_5_0	R	0h	LFSB: Channel causing LFS (Last Frame Started) Interrupt Group B. Contains the channel number of the pending interrupt for group B if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer to below table FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | Channel 31 FTCA 0x20

7.2.1.51 HBCBOFFSET Register (Offset = 168h) [reset = 0h]

HBCBOFFSET is shown in Figure 7-68 and described in Table 7-59.

Return to Summary Table.

Channel causing HBC Interrupt on Group B

Figure 7-68. HBCBOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		HBCB_5_0					
R-0h								R-0h		R-0h					

Table 7-59. HBCBOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	HBCB_5_0	R	0h	0HBCB: Channel causing HBC (Half Block Complete) Interrupt Group B. Contains the channel number of the pending interrupt for group B if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer to below table FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | Channel 31 FTCA 0x20

7.2.1.52 BTCBOFFSET Register (Offset = 16Ch) [reset = 0h]

BTCBOFFSET is shown in Figure 7-69 and described in Table 7-60.

Return to Summary Table.

Channel causing BTC Interrupt on Group B

Figure 7-69. BTCBOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		BTCB_5_0					
R-0h								R-0h		R-0h					

Table 7-60. BTCBOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	BTCB_5_0	R	0h	BTCB: Channel causing BTC (Block Transfer Complete) Interrupt Group B. Contains the channel number of the pending interrupt for group B if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer to below table FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | Channel 31 FTCA 0x20

7.2.1.53 BERBOFFSET Register (Offset = 170h) [reset = 0h]

BERBOFFSET is shown in Figure 7-70 and described in Table 7-61.

Return to Summary Table.

Channel causing BER Interrupt on Group B

Figure 7-70. BERBOFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								sbz		BERB_5_0					
R-0h								R-0h		R-0h					

Table 7-61. BERBOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-6	sbz	R	0h	sbz. These bits should be programmed as zero
5-0	BERB_5_0	R	0h	BERB: Channel causing BER (Bus Error) Interrupt Group B. Contains the channel number of the pending interrupt for group B if the corresponding interrupt enable is set. Reading this location clears the corresponding Interrupt pending flag with the highest priority. Please refer to below table FTCA Interrupt Offset Index Interrupt Condition Offset Value Phantom 0x0 Channel 0 FTCA 0x1 Channel 1 FTCA 0x2 | | | Channel 31 FTCA 0x20

7.2.1.54 PTCRL Register (Offset = 178h) [reset = 0h]

PTCRL is shown in Figure 7-71 and described in Table 7-62.

Return to Summary Table.

Port Control and Status...

Figure 7-71. PTCRL Register

31	30	29	28	27	26	25	24
RESERVED							PENDB
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED					BYB	PSFRHQPB	PSFRLQPB
R-0h					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							PENDA
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED					BYA	PSFRHQPA	PSFRLQPA
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 7-62. PTCRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	PENDB	R	0h	PENDB: Transactions Pending for Port B. This flag determines if there are transactions ongoing on Port B. The flag will be cleared if no transfers are done. It can be used to determine if there is still data transfered while DMA_EN is set to 0 in GCTCRL. In this case, once all transfers are finished, the flag will be set to 0. 1 Transfers are still pending 0 No transfers are pending
23-19	RESERVED	R	0h	Reserved
18	BYB	R/W	0h	BYB: Bypass FIFO B. Writing 1 to this bit limits the FIFO depth to the size of one element. That means after one element is read the write out to the destination will start. This feature is particularly useful to minimize switching latency in-between channels. But it also does not make optimal use of AHB bandwidth.
17	PSFRHQPB	R/W	0h	PSFRHQPB: Priority Scheme Fix or Rotate for High Priority Queue of Port B. 1 Rotation priority 0 Fixed priority
16	PSFRLQPB	R/W	0h	PSFRLQPB: Priority Scheme Fix or Rotate for Low Priority Queue of Port B. 1 Rotation priority 0 Fixed priority
15-9	RESERVED	R	0h	Reserved
8	PENDA	R	0h	PENDA: Transactions Pending for Port A. (Not valid for 16xx) This flag determines if there are transactions ongoing on Port A. The flag will be cleared if no transfers are done. It can be used to determine if there is still data transfered while DMA_EN is set to 0 in GCTCRL. In this case, once all transfers are finished, the flag will be set to 0. 1 Transfers are still pending 0 No transfers are pending
7-3	RESERVED	R	0h	Reserved
2	BYA	R/W	0h	BYA: Bypass FIFO A. (Not valid for 16xx) Writing 1 to this bit limits the FIFO depth to the size of one element. That means after one element is read the write out to the destination will start. This feature is particularly useful to minimize switching latency in-between channels. But it also does not make optimal use of AHB bandwidth
1	PSFRHQPA	R/W	0h	PSFRHQPA: Priority Scheme Fix or Rotate for High Priority Queue of Port A.(Not valid for 16xx) 1 Rotation priority 0 Fixed priority

Table 7-62. PTCRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PSFRLQPA	R/W	0h	PSFRLQPA: Priority Scheme Fix or Rotate for Low Priority Queue of Port A (Not valid for 16xx) 1 Rotation priority 0 Fixed priority

7.2.1.55 RTCTRL Register (Offset = 17Ch) [reset = 0h]

RTCTRL is shown in Figure 7-72 and described in Table 7-63.

Return to Summary Table.

test bit to open write access to CPU during test mode

Figure 7-72. RTCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															RTC
R-0h															R/W-0h

Table 7-63. RTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	RTC	R/W	0h	RTC: RAM Test Control Writing a one to this bit open the write access to the reserved locations of control packet RAM as defined in the memory map. Writing a zero will prohibit write access to the reserved locations of control packet RAM

7.2.1.56 DCTRL Register (Offset = 180h) [reset = 0h]

DCTRL is shown in Figure 7-73 and described in Table 7-64.

Return to Summary Table.

Control bit to enable watchpoint checking

Figure 7-73. DCTRL Register

31	30	29	28	27	26	25	24
RESERVED			CHNUM				
R-0h			R/W-0h				
23	22	21	20	19	18	17	16
RESERVED							DMADBGS
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							DBGEN
R-0h							R/W-0h

Table 7-64. DCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reserved
28-24	CHNUM	R/W	0h	CHNUM : Channel Number. These bit fields indicate the channel number which causes the watchpoint to match
23-17	RESERVED	R	0h	Reserved
16	DMADBGS	R/W	0h	DMADBGS: DMA Debug Status. When a watchpoint is setup to watch for an unique AHB address or a range of addresses is true on one of the three AHB ports is true then DMA Debug Status bit is set and a debug request signal is asserted to the main CPU. CPU must write one to clear this bit in order for DMA to release debug request signal. When this bit is read: 0 = No watchpoint condition is detected. 1 = Watchpoint condition is detected. When this is written: 0 = No effect. 1 = Clears the bit
15-1	RESERVED	R	0h	Reserved
0	DBGEN	R/W	0h	DBGEN: Debug Enable Writing a one to this bit enable the watchpoint checking logics. This bit can only be set during emulation mode when SUSPEND is high. Also this bit is reset by test reset (nTRST).

7.2.1.57 WPR Register (Offset = 184h) [reset = 0h]

WPR is shown in Figure 7-74 and described in Table 7-65.

Return to Summary Table.

watch for an address on AHB bus of either PortA1, PortA2 and PortB

Figure 7-74. WPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WP																															
R/W-0h																															

Table 7-65. WPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WP	R/W	0h	WP: Watch Point Register. This register can only be programmed during emulation mode when SUSPEND signal is high. Also this register is only reset by a test reset (nTRST). A 32 bit address can be programmed into this register as a watchpoint. This register is used along with Watch Mask Register (WMR) as described in the next section. When DBGEN bit is set and a unique address or a range of addresses are detected on the AHB address bus of either Port A1, Port A2 or Port B (Only Port B is valid for 16xx), a debug request signal is sent to ARM CPU. The state machine of the port in which the watchpoint condition is true is frozen

7.2.1.58 WMR Register (Offset = 188h) [reset = 0h]

WMR is shown in Figure 7-75 and described in Table 7-66.

Return to Summary Table.

mask out a bit in watch point register from address compare

Figure 7-75. WMR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WM																															
R/W-0h																															

Table 7-66. WMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WM	R/W	0h	WM: Watch Mask Register. This register can only be programmed during emulation mode when SUSPEND signal is high. Also this register is only reset by a test reset (nTRST). Setting a bit to 1 in the WMR register has the effect of masking the corresponding bit in the WPR register disregarded in the comparison.

7.2.1.59 PAACSADDR Register (Offset = 18Ch) [reset = 0h]

PAACSADDR is shown in Figure 7-76 and described in Table 7-67.

Return to Summary Table.

The current source address of an ongoing active channel in Port A

Figure 7-76. PAACSADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAACSA																															
R/W-0h																															

Table 7-67. PAACSADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PAACSA	R/W	0h	PAACSA: PortA Active Channel Source Address. (Not valid for 16xx) This register contains the current source address of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port A

7.2.1.60 PAACDADDR Register (Offset = 190h) [reset = 0h]

PAACDADDR is shown in Figure 7-77 and described in Table 7-68.

Return to Summary Table.

The destination address of an ongoing active channel in Port A

Figure 7-77. PAACDADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAACDA																															
R/W-0h																															

Table 7-68. PAACDADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PAACDA	R/W	0h	PAACDA: PortA Active Channel Destination Address. (Not valid for 16xx) This register contains the current destination address of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port A.

7.2.1.61 PAACTC Register (Offset = 194h) [reset = 0h]

PAACTC is shown in Figure 7-78 and described in Table 7-69.

Return to Summary Table.

The transfer count of an ongoing active channel in Port B

Figure 7-78. PAACTC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			PAFTCOUNT_28_16												
R-0h			R-0h												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			PAETCOUNT_12_0												
R-0h			R-0h												

Table 7-69. PAACTC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reserved
28-16	PAFTCOUNT_28_16	R	0h	PAFTCOUNT: PortA Active Channel Frame Count. (Not valid for 16xx) This register contains the current frame count value of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port A
15-13	RESERVED	R	0h	Reserved
12-0	PAETCOUNT_12_0	R	0h	PAETCOUNT: PortA Active Channel Element Count. (Not valid for 16xx) This register contains the current element count value of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port A. Note: PAETCOUNT and PAFTCOUNT indicate the count values of the active channel currently residing in DMA's execution queue. With multiple pending channels and due to arbitration between channels the PAETCOUNT and PAFTCOUNT should not be polled to determine the end of a channel transfer.

7.2.1.62 PBACSADDR Register (Offset = 198h) [reset = 0h]

PBACSADDR is shown in Figure 7-79 and described in Table 7-70.

Return to Summary Table.

The current source address of an ongoing active channel in Port B

Figure 7-79. PBACSADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBACSA																															
R/W-0h																															

Table 7-70. PBACSADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PBACSA	R/W	0h	PBACSA: PortB Active Channel Source Address. This register contains the current source address of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port B.

7.2.1.63 PBACDADDR Register (Offset = 19Ch) [reset = 0h]

PBACDADDR is shown in Figure 7-80 and described in Table 7-71.

Return to Summary Table.

The destination address of an ongoing active channel in Port B

Figure 7-80. PBACDADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBACDA																															
R/W-0h																															

Table 7-71. PBACDADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PBACDA	R/W	0h	PBACDA: PortB Active Channel Destination Address. This register contains the current destination address of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port B.

7.2.1.64 PBACTC Register (Offset = 1A0h) [reset = 0h]

PBACTC is shown in Figure 7-81 and described in Table 7-72.

Return to Summary Table.

The transfer count of an ongoing active channel in Port B

Figure 7-81. PBACTC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			PBFTCOUNT_28_16												
R-0h			R/W-0h												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			PBETCOUNT_12_0												
R-0h			R/W-0h												

Table 7-72. PBACTC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reserved
28-16	PBFTCOUNT_28_16	R/W	0h	PBFTCOUNT: PortB Active Channel Frame Count. This register contains the current frame count value of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port B.
15-13	RESERVED	R	0h	Reserved
12-0	PBETCOUNT_12_0	R/W	0h	PBETCOUNT: PortB Active Channel Element Count. This register contains the current element count value of the active channel as broadcasted in Section 1.6.2.4, "DMA STATUS Register (DMASTAT)" for Port B.

7.2.1.65 DMAPCR Register (Offset = 1A8h) [reset = 5h]

DMAPCR is shown in Figure 7-82 and described in Table 7-73.

Return to Summary Table.

Control of the parity generation

Figure 7-82. DMAPCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							ERRA
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED							TEST
R-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED				PARITY_ENA			
R-0h				R/W-5h			

Table 7-73. DMAPCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	Reserved
16	ERRA	R/W	0h	ERRA: Error Action 0 = If a parity error is detected on Control Packet x (x = 0, 1, ...or 31), then the enable/disable state of Control Packet x remains unchanged. 1 = If a parity error is detected on Control Packet x (x = 0, 1, ...or 31), then DMA is disabled immediately. If a frame on Control Packet x is processed at the time the parity error is detected, then remaining elements of this frame will no more be transferred. Disabling of the DMA will be done, regardless of whether the error was detected during a read to the Control Packet RAM performed by the DMA state machine or by a different master.
15-9	RESERVED	R	0h	Reserved
8	TEST	R/W	0h	TEST. When this bit is set, the parity bits are memory mapped to make them accessible by the CPU. 0 = parity bits are not memory mapped 1 = parity bits are memory mapped
7-4	RESERVED	R	0h	Reserved
3-0	PARITY_ENA	R/W	5h	PARITY_ENA: Parity Error Detection Enable. This bit field enables or disables the parity check on read operations and the parity calculation on write operations. If parity checking is enabled and a parity error is detected the DMA_UERR signal is activated. During memory initiallization this bit field needs to be enable inorder to generate parity bits inside memory. 0101 = disable all other = enable NOTE: It is recommended to write a ‘1010’ to enable error detection, to guard against soft error from flipping PARITY_ENA to a disable state

7.2.1.66 DMAPAR Register (Offset = 1ACh) [reset = 0h]

DMAPAR is shown in Figure 7-83 and described in Table 7-74.

Return to Summary Table.

The address of the control packet memory, where the uncorrectable error occured

Figure 7-83. DMAPAR Register

31	30	29	28	27	26	25	24
RESERVED							EDFLG
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				ERROR ADDRESS_11_0			
R-0h				R-0h			
7	6	5	4	3	2	1	0
ERROR ADDRESS_11_0							
R-0h							

Table 7-74. DMAPAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	EDFLG	R	0h	EDFLG: Parity Error Detection Flag. This flag indicates if a parity error occured on reading DMA Control packet RAM. When this bit is read: 0 = No error occured. 1 = Error detected and the address is captured in DMAPAR's ERROR_ADDRESS field. When write to this bit with: 0 = No effect. 1 = Clears the bit.
23-12	RESERVED	R	0h	Reserved
11-0	ERROR ADDRESS_11_0	R	0h	ERROR ADDRESS: Error Address. This register holds the address of the first parity error generated in the RAM. This error address is frozen from being updated until it is read by the CPU. During emulation mode when SUSPEND is high, this address is frozen even when read. NOTE: The Error Address Register (11:0) will not be reset, neither by PORRST nor by any other reset source.

7.2.1.67 DMAMPCTRL Register (Offset = 1B0h) [reset = 0h]

DMAMPCTRL is shown in Figure 7-84 and described in Table 7-75.

Return to Summary Table.

Controls enablin/disabling of the protection regions and access restrictions

Figure 7-84. DMAMPCTRL Register

31	30	29	28	27	26	25	24
RESERVED			INT3AB	INT3ENA	REG3AP		REG3ENA
R-0h			R/W-0h	R/W-0h	R/W-0h		R/W-0h
23	22	21	20	19	18	17	16
RESERVED			INT2AB	INT2ENA	REG2AP		REG2ENA
R-0h			R/W-0h	R/W-0h	R/W-0h		R/W-0h
15	14	13	12	11	10	9	8
RESERVED			INT1AB	INT1ENA	REG1AP		REG1ENA
R-0h			R/W-0h	R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
RESERVED			INT0AB	INT0ENA	REG0AP		REG0ENA
R-0h			R/W-0h	R/W-0h	R/W-0h		R/W-0h

Table 7-75. DMAMPCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reserved
28	INT3AB	R/W	0h	INT3AB: Interrupt assignment of region 3 to Group A or GroupB. 0 = Interrupt is routed to VIM (Group A) 1 = Interrupt is routed to DSP CPU (Group B) (Not valid for 16xx)
27	INT3ENA	R/W	0h	INT3ENA: Interrupt enable of region 3. 0 = Interrupt is disabled 1 = Interrupt is enabled
26-25	REG3AP	R/W	0h	REG3AP: Region 3 Access Permission. These bits determine the access permission for region 3 REG3AP Access Permission 00 all accesses allowed 01 read only 10 write only 11 none
24	REG3ENA	R/W	0h	REG3ENA: Region 3 Enable. 0 = Region is disabled (no address checking done) 1 = Region is enalbed (address and access permission checking done)
23-21	RESERVED	R	0h	Reserved
20	INT2AB	R/W	0h	INT2AB: Interrupt assignment of region 2 to Group A or GroupB. 0 = Interrupt is routed to VIM (Group A) 1 = Interrupt is routed to DSP CPU (Group B) (Not valid for 16xx)
19	INT2ENA	R/W	0h	INT2ENA: Interrupt enable of region 2. 0 = Interrupt is disabled 1 = Interrupt is enabled
18-17	REG2AP	R/W	0h	REG2AP: Region 2 Access Permission. These bits determine the access permission for region 2 REG2AP Access Permission 00 all accesses allowed 01 read only 10 write only 11 none
16	REG2ENA	R/W	0h	REG2ENA: Region 2 Enable. 0 = Region is disabled (no address checking done) 1 = Region is enalbed (address and access permission checking done)
15-13	RESERVED	R	0h	Reserved
12	INT1AB	R/W	0h	INT1AB: Interrupt assignment of region 1 to Group A or GroupB. 0 = Interrupt is routed to VIM (Group A) 1 = Interrupt is routed to DSP CPU (Group B) (Not valid for 16xx)
11	INT1ENA	R/W	0h	INT1ENA: Interrupt enable of region 1. 0 = Interrupt is disabled 1 = Interrupt is enabled

Table 7-75. DMAMPCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-9	REG1AP	R/W	0h	REG1AP: Region 1 Access Permission These bits determine the access permission for region 1 REG1AP Access Permission 00 all accesses allowed 01 read only 10 write only 11 none
8	REG1ENA	R/W	0h	REG1ENA: Region 1 Enable. 0 = Region is disabled (no address checking done) 1 = Region is enalbed (address and access permission checking done)
7-5	RESERVED	R	0h	Reserved
4	INT0AB	R/W	0h	INT0AB: Interrupt assignment of region 0 to Group A or GroupB. 0 = Interrupt is routed to VIM (Group A) 1 = Interrupt is routed to DSP CPU (Group B) (Not valid for 16xx)
3	INT0ENA	R/W	0h	INT0ENA: Interrupt enable of region 0. 0 = Interrupt is disabled 1 = Interrupt is enabled
2-1	REG0AP	R/W	0h	REG0AP: Region 0 Access Permission. These bits determine the access permission for region 0 REG0AP Access Permission 00 all accesses allowed 01 read only 10 write only 11 none
0	REG0ENA	R/W	0h	REG0ENA: Region 0 Enable. 0 = Region is disabled (no address checking done) 1 = Region is enalbed (address and access permission checking done)

7.2.1.68 DMAMPST Register (Offset = 1B4h) [reset = 0h]

DMAMPST is shown in Figure 7-85 and described in Table 7-76.

Return to Summary Table.

Status bits to show in which region a access violation happened

Figure 7-85. DMAMPST Register

31	30	29	28	27	26	25	24
RESERVED							REG3FT
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							REG2FT
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							REG1FT
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							REG0FT
R-0h							R-0h

Table 7-76. DMAMPST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	REG3FT	R	0h	REG3FT: Region 3 Fault. This bit determines whether a access permission violation was detected in this region. Once the bit is set, it can be cleared by writing a 1. 0 = no fault detected 1 = fault detected
23-17	RESERVED	R	0h	Reserved
16	REG2FT	R	0h	REG2FT: Region 2 Fault. This bit determines whether a access permission violation was detected in this region. Once the bit is set, it can be cleared by writing a 1. 0 = no fault detected 1 = fault detected
15-9	RESERVED	R	0h	Reserved
8	REG1FT	R	0h	REG1FT: Region 1 Fault. This bit determines whether a access permission violation was detected in this region. Once the bit is set, it can be cleared by writing a 1. 0 = no fault detected 1 = fault detected
7-1	RESERVED	R	0h	Reserved
0	REG0FT	R	0h	REG0FT: Region 0 Fault. This bit determines whether a access permission violation was detected in this region. Once the bit is set, it can be cleared by writing a 1. 0 = no fault detected 1 = fault detected

7.2.1.69 DMAMPR0S Register (Offset = 1B8h) [reset = 0h]

DMAMPR0S is shown in Figure 7-86 and described in Table 7-77.

Return to Summary Table.

Defines starting address of region 0

Figure 7-86. DMAMPR0S Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STARTADDRESS																															
R/W-0h																															

Table 7-77. DMAMPR0S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STARTADDRESS	R/W	0h	STARTADDRESS - Region 0 The startaddress defines at which address the region begins

7.2.1.70 DMAMPR0E Register (Offset = 1BCh) [reset = 0h]

DMAMPR0E is shown in Figure 7-87 and described in Table 7-78.

Return to Summary Table.

Defines end address of region 0

Figure 7-87. DMAMPR0E Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDADDRESS																															
R/W-0h																															

Table 7-78. DMAMPR0E Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENDADDRESS	R/W	0h	ENDADDRESS - Region 0 The endaddress defines at which address the region ends. The endaddress usually is bigger than the startaddress for this region. Otherwise the region will wrap around at the end of the address space.

7.2.1.71 DMAMPR1S Register (Offset = 1C0h) [reset = 0h]

DMAMPR1S is shown in Figure 7-88 and described in Table 7-79.

Return to Summary Table.

Defines starting address of region 1

Figure 7-88. DMAMPR1S Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STARTADDRESS																															
R/W-0h																															

Table 7-79. DMAMPR1S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STARTADDRESS	R/W	0h	STARTADDRESS - Region 1 The startaddress defines at which address the region begins

7.2.1.72 DMAMPR1E Register (Offset = 1C4h) [reset = 0h]

DMAMPR1E is shown in Figure 7-89 and described in Table 7-80.

Return to Summary Table.

Defines end address of region 1

Figure 7-89. DMAMPR1E Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDADDRESS																															
R/W-0h																															

Table 7-80. DMAMPR1E Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENDADDRESS	R/W	0h	ENDADDRESS - Region 1 The endaddress defines at which address the region ends. The endaddress usually is bigger than the startaddress for this region. Otherwise the region will wrap around at the end of the address space.

7.2.1.73 DMAMPR2S Register (Offset = 1C8h) [reset = 0h]

DMAMPR2S is shown in Figure 7-90 and described in Table 7-81.

Return to Summary Table.

Defines starting address of region 2

Figure 7-90. DMAMPR2S Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STARTADDRESS																															
R/W-0h																															

Table 7-81. DMAMPR2S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STARTADDRESS	R/W	0h	STARTADDRESS - Region 2 The startaddress defines at which address the region begins.

7.2.1.74 DMAMPR2E Register (Offset = 1CCh) [reset = 0h]

DMAMPR2E is shown in Figure 7-91 and described in Table 7-82.

Return to Summary Table.

Defines end address of region 2

Figure 7-91. DMAMPR2E Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDADDRESS																															
R/W-0h																															

Table 7-82. DMAMPR2E Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENDADDRESS	R/W	0h	ENDADDRESS - Region 2 The endaddress defines at which address the region ends. The endaddress usually is bigger than the startaddress for this region. Otherwise the region will wrap around at the end of the address space.

7.2.1.75 DMAMPR3S Register (Offset = 1D0h) [reset = 0h]

DMAMPR3S is shown in Figure 7-92 and described in Table 7-83.

Return to Summary Table.

Defines starting address of region 3

Figure 7-92. DMAMPR3S Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STARTADDRESS																															
R/W-0h																															

Table 7-83. DMAMPR3S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STARTADDRESS	R/W	0h	STARTADDRESS - Region 3 The startaddress defines at which address the region begins.

7.2.1.76 DMAMPR3E Register (Offset = 1D4h) [reset = 0h]

DMAMPR3E is shown in Figure 7-93 and described in Table 7-84.

Return to Summary Table.

Defines end address of region 3

Figure 7-93. DMAMPR3E Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDADDRESS																															
R/W-0h																															

Table 7-84. DMAMPR3E Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENDADDRESS	R/W	0h	ENDADDRESS - Region 3 The endaddress defines at which address the region ends. The endaddress usually is bigger than the startaddress for this region. Otherwise the region will wrap around at the end of the address space.

7.2.2 Channel Configuration

The channel configuration is defined by the channel control packet: channel control, transfer count, index pointers, source/destination address.

- It is stored in local RAM, which is protected by parity.
- Each control packet contains a total of nine fields.
- The first six fields are programmable, while the last three fields are read only.
- The RAM is accessible by queue A and queue B state machines as well as CPU.
- When there are simultaneous accesses, the priority is resolved in a fixed priority scheme with the CPU having the highest priority.

All the control packets look the same. Following, there is the detailed layout of these registers shown for control packet 0.

7.2.2.1 Initial Source Address (ISADDR)

Figure 7-94. Initial Source Address (ISADDR) [offset = 00]

31	16
ISADDR	
R/WP-X	

15	0
ISADDR	
R/WP-X	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset; X = Unknown

Table 7-85. Initial Source Address (ISADDR) Field Descriptions

Bit	Field	Value	Description
31-0	ISADDR	0-FFFF FFFFh	Initial source address. These bits give the absolute 32-bit source address (physical).

7.2.2.2 Initial Destination Address Register (IDADDR)

Figure 7-95. Initial Destination Address Register (IDADDR) [offset = 04h]

31	16
IDADDR	
R/WP-X	

15	0
IDADDR	
R/WP-X	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset; X = Unknown

Table 7-86. Initial Destination Address Register (IDADDR) Field Descriptions

Bit	Field	Value	Description
31-0	IDADDR	0-FFFF FFFFh	Initial destination address. These bits give the absolute 32-bit destination address (physical).

7.2.2.3 Initial Transfer Count Register (ITCOUNT)

Figure 7-96. Initial Transfer Count Register (ITCOUNT) [offset = 08h]

31	29	28	16
Reserved		IFTCOUNT	
R-X		R/WP-X	

15	13	12	0
Reserved		IETCOUNT	
R-X		R/WP-X	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset; X = Unknown

Table 7-87. Initial Transfer Count Register (ITCOUNT) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reads are undefined. Writes have no effect.
28-16	IFTCOUNT	0-1FFFh	Initial frame transfer count. These bits define the number of frame transfers.
15-13	Reserved	0	Reads are undefined. Writes have no effect.
12-0	IETCOUNT	0-1FFFh	Initial element transfer count. These bits define the number of element transfers. The block transfer size will be IETCOUNT x IFTCOUNT

7.2.2.4 Channel Control Register (CHCTRL)

Figure 7-97. Channel Control Register (CHCTRL) [offset = 10h]

31	22	21	16
Reserved		CHAIN	
R-X		R/WP-X	

15	14	13	12	11	9	8	7	5	4	3	2	1	0
RES		WES		Reserved		TTYPE	Reserved		ADDMR		ADDMW		AIM
R/WP-X		R/WP-X		R-X		R/WP-X	R-X		R/WP-X		R/WP-X		R/WP-X

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset; X = Unknown

Table 7-88. Channel Control Register (CHCTRL) Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reads are undefined. Writes have no effect.
21-16	CHAIN		Next channel to be triggered. At the end of the programmed number of frames, the specified channel will be triggered. Programmed value x means channel (x-1) is chained.
			Note: The programmer must program the CHAIN bits before initiating a DMA transfer.
		0	No channel is selected.
		1h	Channel 0 is selected.
		:	:
		10h	Channel 15 is selected.
		11h-3Fh	Reserved
15-14	RES		Read element size.
		0	The element is byte, 8-bit.
		1h	The element is half-word, 16-bit.
		2h	The element is word, 32-bit.
		3h	The element is double-word, 64-bit.
13-12	WES		Write element size.
		0	The element is byte, 8-bit.
		1h	The element is half-word, 16-bit.
		2h	The element is word, 32-bit.
		3h	The element is double-word, 64-bit.
11-9	Reserved	0	Reads are undefined. Writes have no effect.
8	TTYPE		Transfer type.
		0	A hardware request triggers one frame transfer.
		1	A hardware request triggers one block transfer.
7-5	Reserved	0	Reads are undefined. Writes have no effect.
4-3	ADDMR		Addressing mode read.
		0	Constant
		1h	Post-increment
		2h	Reserved
		3h	Indexed
2-1	ADDMW		Addressing mode write.
		0	Constant
		1h	Post-increment
		2h	Reserved
		3h	Indexed
0	AIM		Auto-initiation mode.
		0	Auto-initiation mode is disabled.
		1	Auto-initiation mode is enabled.

7.2.2.5 Element Index Offset Register (EIOFF)

Figure 7-98. Element Index Offset Register (EIOFF) [offset = 14h]

31 29	28 16
Reserved	EIDXD
R-X	R/WP-X

15 13	12 0
Reserved	EIDXS
R-X	R/WP-X

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset; X = Unknown

Table 7-89. Element Index Offset Register (EIOFF) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reads are undefined. Writes have no effect.
28-16	EIDXD	0-1FFFh	Destination address element index. These bits define the offset to be added to the destination address after each element transfer.
15-13	Reserved	0	Reads are undefined. Writes have no effect.
12-0	EIDXS	0-1FFFh	Source address element index. These bits define the offset to be added to the source address after each element transfer.

7.2.2.6 Frame Index Offset Register (FIOFF)

Figure 7-99. Frame Index Offset Register (FIOFF) [offset = 18h]

31 29	28 16
Reserved	FIDXD
R-X	R/WP-X

15 13	12 0
Reserved	FIDXS
R-X	R/WP-X

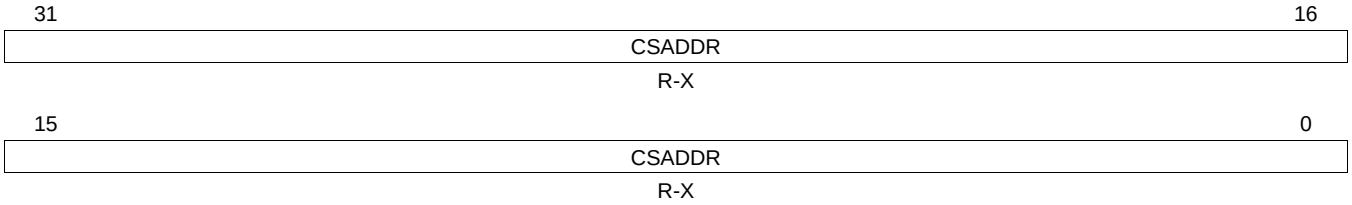
LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset; X = Unknown

Table 7-90. Frame Index Offset Register (FIOFF) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reads are undefined. Writes have no effect.
28-16	FIDXD	0-1FFFh	Destination address frame index. These bits define the offset to be added to the destination address after element count reached 1.
15-13	Reserved	0	Reads are undefined. Writes have no effect.
12-0	FIDXS	0-1FFFh	Source address frame index. These bits define the offset to be added to the source address after element count reached 1.

7.2.2.7 Current Source Address Register (CSADDR)

Figure 7-100. Current Source Address Register (CSADDR) [offset = 800h]



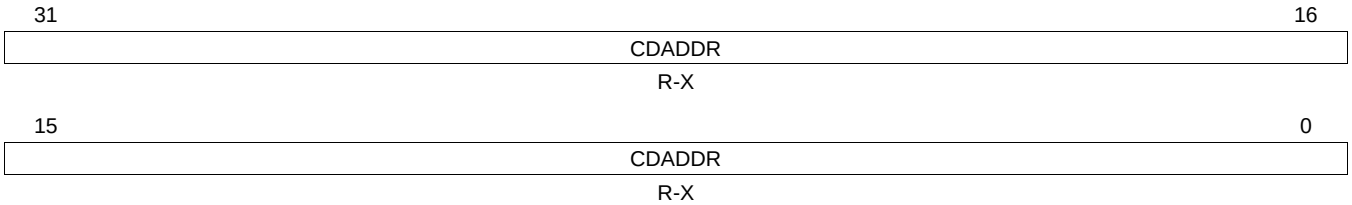
LEGEND: R = Read only; -*n* = value after reset; X = Unknown

Table 7-91. Current Source Address Register (CSADDR) Field Descriptions

Bit	Field	Value	Description
31-0	CSADDR	0-FFFF FFFFh	Current source address. These bits contain the current working absolute 32-bit source address (physical). These bits are only updated after a channel is arbitrated out from the priority queue.

7.2.2.8 Current Destination Address Register (CDADDR)

Figure 7-101. Current Destination Address Register (CDADDR) [offset = 804h]



LEGEND: R = Read only; -*n* = value after reset; X = Unknown

Table 7-92. Current Destination Address Register (CDADDR) Field Descriptions

Bit	Field	Value	Description
31-0	CDADDR	0-FFFF FFFFh	Current destination address. These bits contain the current working absolute 32-bit destination address (physical). These bits are only updated after a channel is arbitrated out of the priority queue.

7.2.2.9 Current Transfer Count Register (CTCOUNT)

Figure 7-102. Current Transfer Count Register (CTCOUNT) [offset = 808h]

31	29	28	16
Reserved		CFTCOUNT	
R-X		R-X	

15	13	12	0
Reserved		CETCOUNT	
R-X		R-X	

LEGEND: R = Read only; -*n* = value after reset; X = Unknown

Table 7-93. Current Transfer Count Register (CTCOUNT) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reads are undefined. Writes have no effect.
28-16	CFTCOUNT	0-1FFFh	Current frame transfer count. Returned the current remaining frame counts.
15-13	Reserved	0	Reads are undefined. Writes have no effect.
12-0	CETCOUNT	0-1FFFh	Current element transfer count. These bits return the current remaining element counts. CTCOUNT register is only updated after a channel is arbitrated out of the priority queue.

7.3 DMA Request Map

For a the device-specific DMA request maps, refer to Table 1-8 for the 14xx and Table 3-10 for the 16xx/18xx/68xx devices.

Vectored Interrupt Manager (VIM)

This chapter describes the behavior of the vectored interrupt manager (VIM) module.

Topic **Page**

8.1 Overview 1510
8.2 Device Level Interrupt Management 1511
8.3 Interrupt Handling Inside VIM 1514
8.4 Interrupt Vector Table (VIM RAM) 1518
8.5 VIM Wakeup Interrupt 1523
8.6 Capture Event Sources 1524
8.7 Examples 1524
8.8 VIM Control Registers 1527
8.9 Interrupt Request Assignments 1542

8.1 Overview

The vectored interrupt manager (VIM) provides hardware assistance for prioritizing and controlling the many interrupt sources present on a device. Interrupts are caused by events outside of the normal flow of program execution. Normally, these events require a timely response from the central processing unit (CPU); therefore, when an interrupt occurs, the CPU switches execution from the normal program flow to an interrupt service routine (ISR).

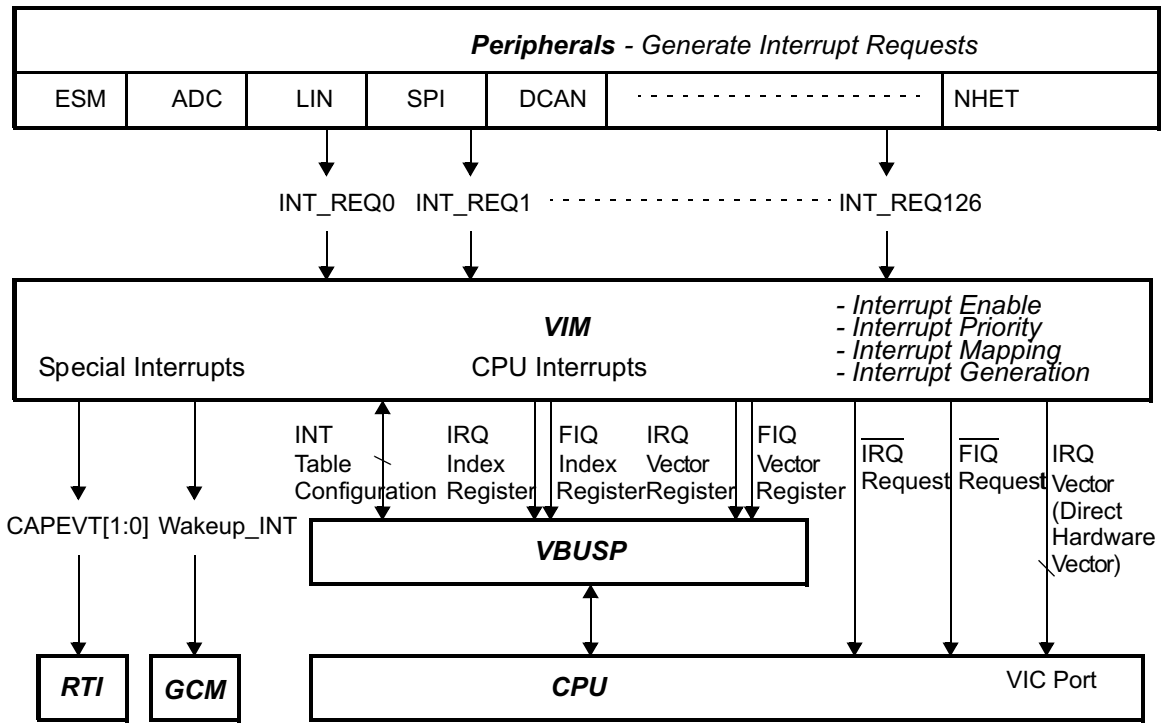
The VIM module has the following features:

- Supports 127 interrupt channels, in both register vectored interrupt and hardware vectored interrupt mode.
  - Provides IRQ vector directly to the CPU VIC port
  - Provides FIQ/IRQ vector through registers
  - Provides programmable priority and enable for interrupt request lines
- Provides a direct hardware dispatch mechanism for fastest IRQ dispatch.
- Provides two software dispatch mechanisms for backward compatibility with earlier generation of TI processors.
  - Index interrupt
  - Register vectored interrupt
- ECC (Error Code Correction) protected vector interrupt table against soft errors.

8.2 Device Level Interrupt Management

A block diagram of device level interrupt handling is shown in Figure 8-1. When an event occurs within a peripheral, the peripheral makes an interrupt request to the VIM. Then, VIM prioritizes the requests from peripherals and provides the address of the highest interrupt service routine (ISR) to the CPU. Finally, CPU starts executing the ISR instructions from that address in the ISR. Section 8.2.1 through Section 8.2.3 provide additional details about these three steps.

Figure 8-1. Device Level Interrupt Block Diagram



8.2.1 Interrupt Generation at the Peripheral

Interrupt generation begins when an event occurs within a peripheral module. Some examples of interrupt-capable events are expiration of a counter within a timer module, receipt of a character in a communications module, and completion of a conversion in an analog-to-digital converter (ADC) module. Some device peripherals are capable of requesting interrupts on more than one interrupt request line.

Interrupts are not always generated when an event occurs; the peripheral must make an interrupt request to the VIM based on the event occurrence. Typically, the peripheral contains:

- An interrupt flag bit for each event to signify the event occurrence.
- An interrupt enable bit to control whether the event occurrence causes an interrupt request to the VIM.

8.2.2 Interrupt Handling at the CPU

The ARM CPU provides two vectors for interrupt requests—fast interrupt requests (FIQs) and normal interrupt requests (IRQs). FIQs are higher priority than IRQs, and FIQ interrupts may interrupt IRQ interrupts.

NOTE: The FIQ implemented in Cortex-R4F/R5F is Non-Maskable Fast Interrupts (NMFI). Once FIQ is enabled (by clearing F bit in CPSR), it can NOT be disabled by setting F bit in CPSR. Only a reset or an FIQ will be able to set the F bit in CPSR. By hardware, Non Maskable FIQ are not reentrant.

After reset (power reset or warm reset), both FIQ and IRQ are disabled. The CPU may enable these interrupt request channels individually within the CPSR (Current Program Status Register); CPSR bits 6 and 7 must be cleared to enable the FIQ (bit 6) and IRQ (bit 7) interrupt requests at the CPU. CPSR is writable in privilege mode only. Example 8-2 shows how to enable the IRQ and FIQ through CPSR.

When the CPU receives an interrupt request, the CPSR mode field changes to either FIQ or IRQ mode. When an IRQ interrupt is received, the CPU disables other IRQ interrupts by setting CPSR bit 7. When an FIQ interrupt is received, the CPU disables both IRQ and FIQ interrupts by setting CPSR bits 6 and 7.

A write of 1 to CPSR bit 7 disables the IRQ from CPU. However, a write of 1 to CPSR bit 6 leaves it unchanged. Example 8-2 also shows how to disable the IRQ through CPSR.

8.2.3 Software Interrupt Handling Options

The device supports three different possibilities for software to handle interrupts

1. Index interrupts mode

   After the interrupt is received by the CPU, the CPU branches to 0x18 (IRQ) or 0x1C (FIQ) to execute the main ISR. The main ISR routine reads the offset register (IRQINDEX, FIQINDEX) to determine the source of the interrupt.

2. Register vectored interrupts (automatically provide vector address to application)

   Before enabling interrupts, the application software also has to initiate the interrupt vector table (VIM RAM).

   Once the VIM receives an interrupt, it loads the address of ISR from interrupt vector table, and store it into the interrupt vector register (IRQVECREG for IRQ interrupt, FIQVECREG for FIQ interrupt).

   After the interrupt is received by the CPU, the CPU executes the instruction placed at 0x18 or 0x1C (IRQ or FIQ vector) to load the address of ISR (interrupt vector) from the interrupt vector register. Example 8-3 illustrates the configuration for the exception vectors using this mode.

3. Hardware vectored interrupts (automatically dispatch to ISR, IRQ only)

   Before enabling interrupts, the application software must initiate the interrupt vector table (VIM RAM) pointing to the ISR for each interrupt channel.

   After the interrupt (IRQ) is received by the CPU, CPU reads the address of ISR directly from the interface with VIM (VIC port) instead of branching to 0x18. The CPU will branch directly to the ISR.

   The hardware vectored interrupt behavior must be explicitly enabled by setting the vector enable (VE) bit in the CP15 R1 register. This bit resets to 0, so that the default state after reset is backward compatible to earlier ARM CPU. Example 8-1 shows how to enable the hardware vectored interrupt.

   NOTE: This mode is NOT available for FIQ.

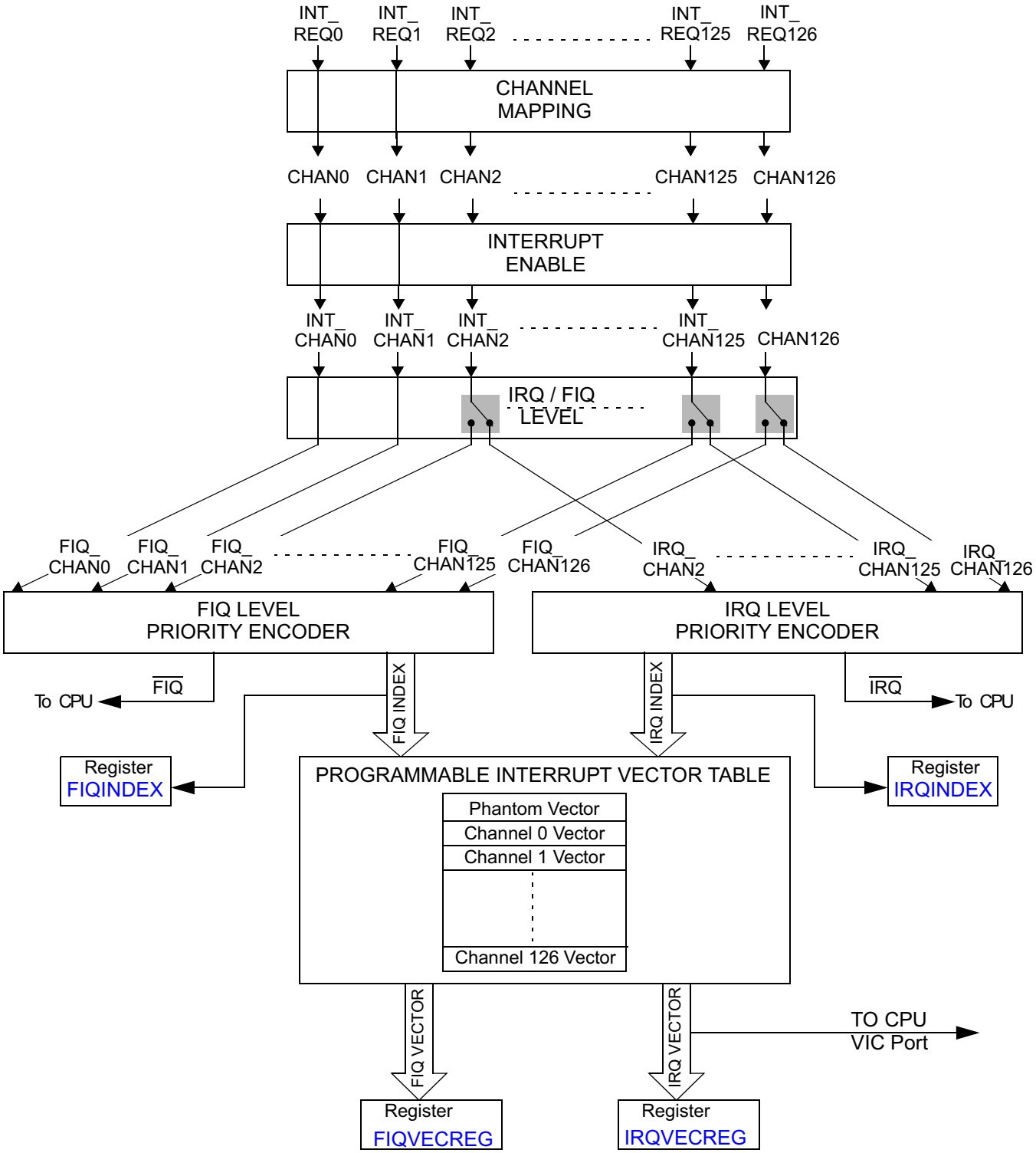
4. Software-Based Priority Decoding Scheme

   If the application uses a software-based interrupt priority decoding scheme instead of the hardware vector capabilities, then there is an additional step which was not required on earlier devices. This version of the VIM will hold an interrupt request generated by a peripheral. When the software clears the interrupt condition in the source module (for example, RTI, GIO, and so on), then it must also perform an additional clear of the interrupt request in the VIM. This can be done by reading the IRQVECREG register (Section 8.8.15) or FIQVECREG register (Section 8.8.16), or by writing a 1 to the INTREQ(i) bit (Section 8.8.10) in the VIM. This is not necessary if any of the three previous methods are used as the interrupt request bit in the VIM will be automatically cleared when the vector is read.

8.3 Interrupt Handling Inside VIM

A block diagram of the interrupt handling inside VIM is shown in Figure 8-2

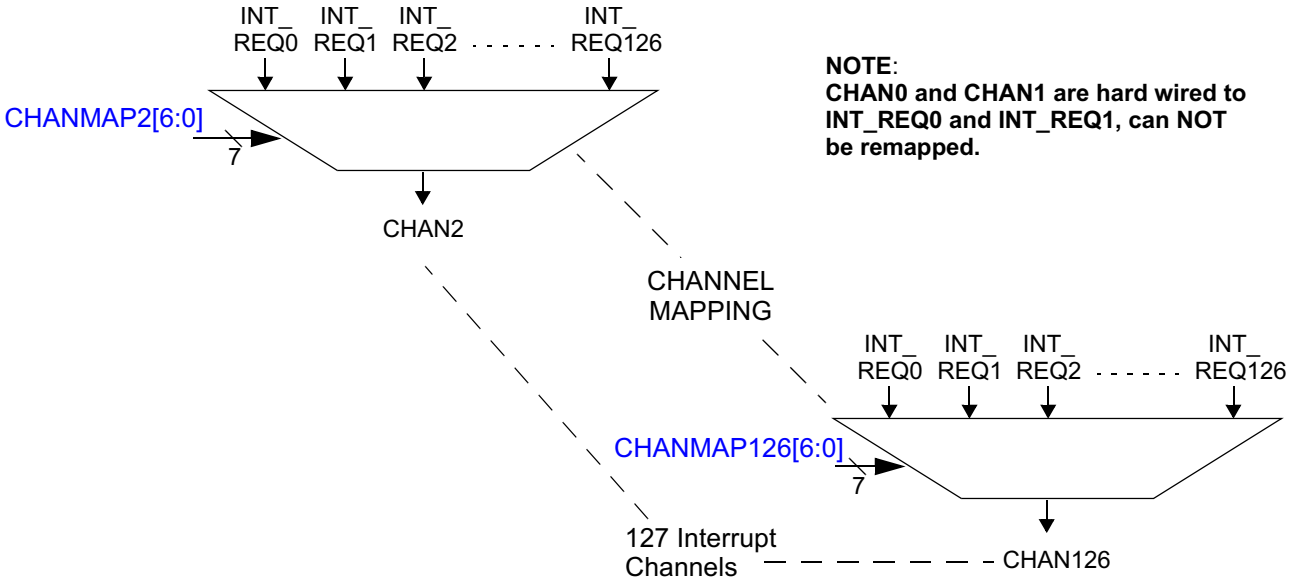
Figure 8-2. VIM Interrupt Handling Block Diagram



8.3.1 VIM Interrupt Channel Mapping

The VIM support 128 interrupt channels (including phantom interrupt). A block diagram of the VIM interrupt requests arrangement from peripheral modules to the interrupt channels is provided in Figure 8-3. Each interrupt channel (CHANx) has a corresponding mapping register bit field (CHANMAPx[6:0]). This mapping register determines which interrupt channel it maps each VIM interrupt request. With this scheme, the same request can be mapped to multiple channels. A lower numbered channel in each FIQ and IRQ has higher priority. The programmability of the VIM allows software to control the interrupt priority.

Figure 8-3. VIM Channel Mapping



NOTE: CHAN127

CHAN127 has no dedicated interrupt vector table entry. Therefore, CHAN127 shall NOT be remapped to other INT_REQ (INT_REQ127 is reserved at device level).

In the reset state, the VIM maps all of the interrupt requests in the system to their respective interrupt channels. Figure 8-4 shows the default state following the reset.

Figure 8-5 shows the VIM INT2 is remapped to both Channel 2 and 4, and INT3 is mapped to channel 3.

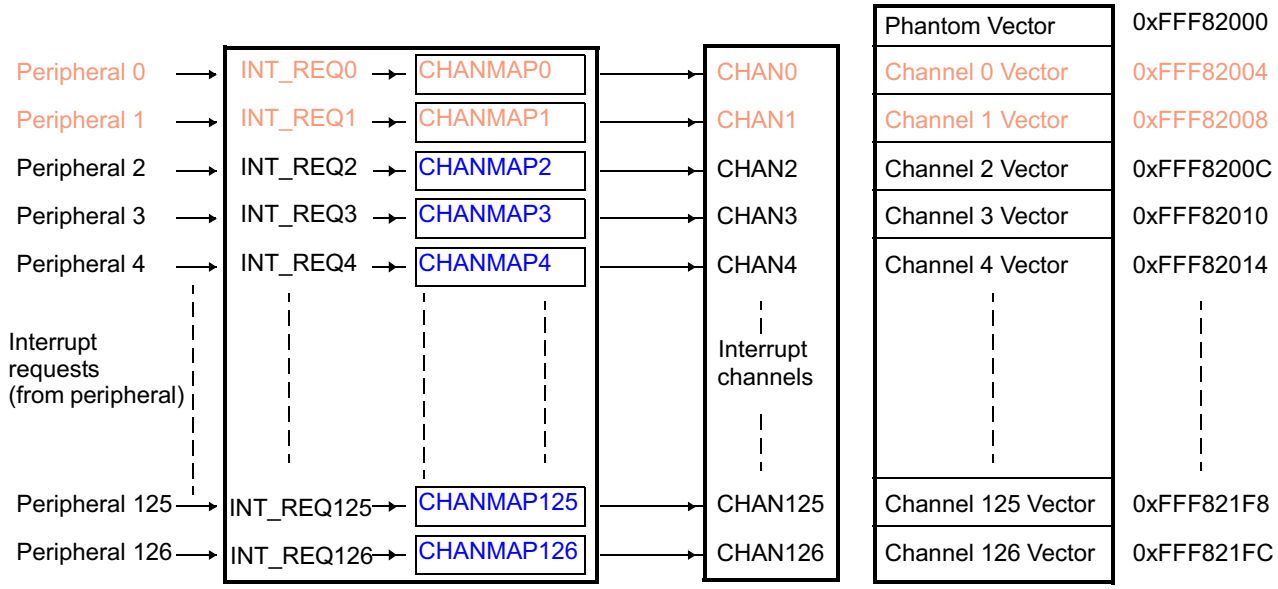
NOTE: By mapping INT2 to channel 2 and channel 4, and mapping INT3 to channel 3, it is possible for the software to change the priority dynamically by changing the ENABLE register (REQENASET and REQENACLR). When channel 2 is enabled, the priority is:

1. INT0
2. INT1
3. INT2
4. INT3

Disabling channel 2, the priority becomes:

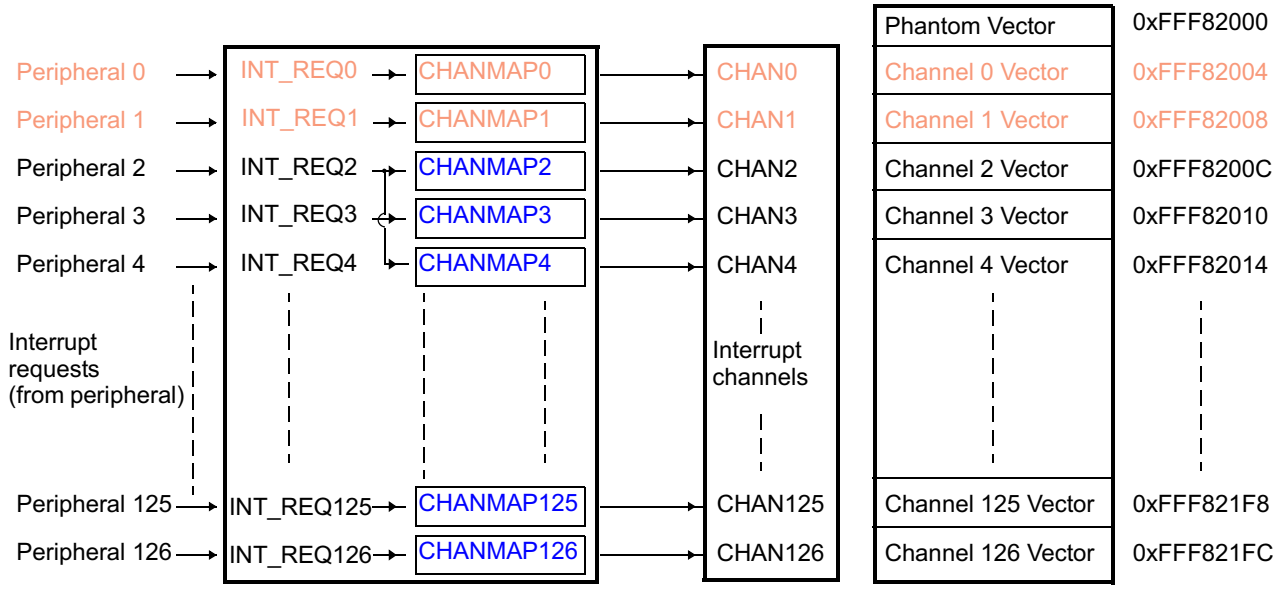
1. INT0
2. INT1
3. INT3
4. INT2

Figure 8-4. VIM in Default State



NOTE: CHAN0 and CHAN1 are hardwired to INT_REQ0 and INT_REQ1, so they cannot be remapped.

Figure 8-5. VIM in a Programmed State



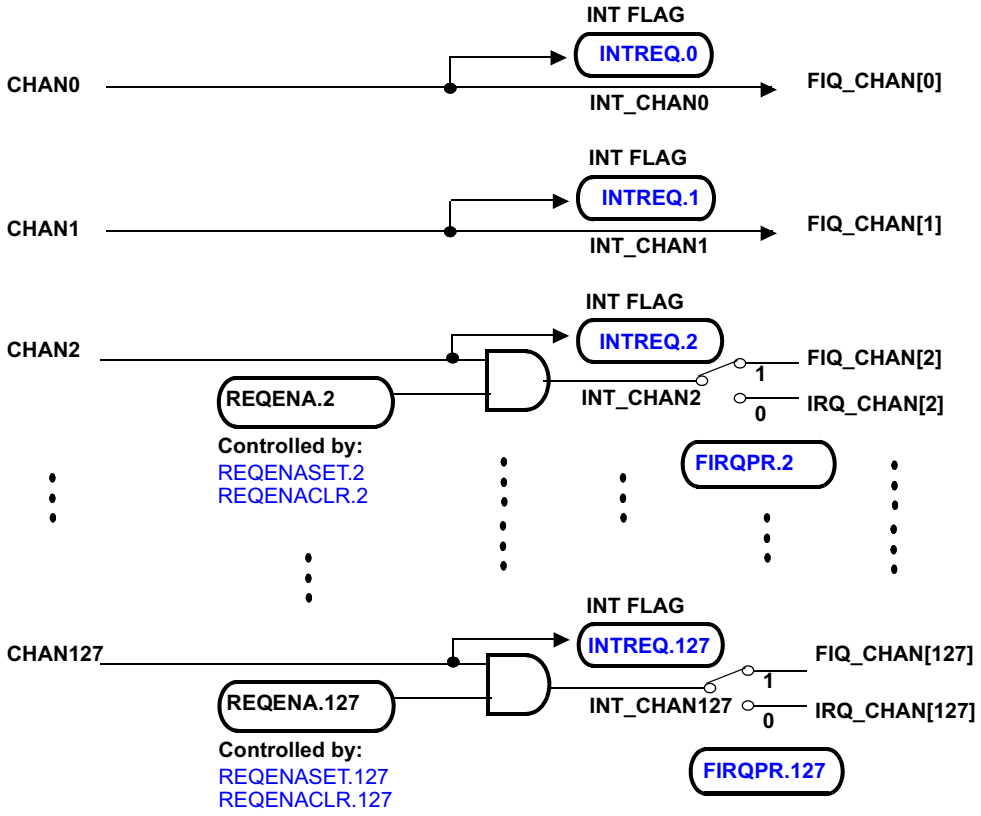
NOTE: CHAN0 and CHAN1 are hard wired to INT_REQ0 and INT_REQ1, so they cannot be remapped.

8.3.2 VIM Input Channel Management

As shown in Figure 8-6, the VIM enables channels on a channel-by-channel basis (in the REQENASET and REQENACLR registers); unused channels may be masked to prevent spurious interrupts.

NOTE: The interrupt ENABLE register does not affect the value of INTREQ.

Figure 8-6. Interrupt Channel Management



By default, interrupt CHAN0 is mapped to ESM (Error Signal Module) high level interrupt and CHAN1 is reserved for other NMI. For safety reasons, these two channels are mapped to FIQ only and can **NOT** be disabled through ENABLE registers.

NOTE: NMI Channel

Channel 0 and channel 1 are not maskable by the REQENASET / REQENACLR bit and both channel are routed exclusively to FIQ/NMI request line (FIRQPR0 and FIRQPR1 have no effect).

The VIM prioritizes the received interrupts based upon a programmed prioritization scheme. The VIM can send two interrupt requests to the CPU simultaneously—one IRQ and one FIQ. If both interrupt types are enabled at the CPU level, then the FIQ has greater priority and is handled first. Each interrupt channel, except channel 0 and 1, can be assigned to send either an FIQ or IRQ request to the CPU (in the FIRQPR register).

The VIM provides a default prioritization scheme, which sends the lowest numbered active channel (in each FIQ and IRQ classes) to the CPU. Within the FIQ and IRQ classes of interrupts, the lowest channel has the highest priority interrupt. The channel number is programmable through register CHANMAPx.

After the VIM has generated the vector corresponding to the highest active IRQ, it updates the FIQINDEX or the IRQINDEX register, depending on the class of interrupt. Then, it accesses the interrupt vector table using the vector value to fetch the address of the corresponding ISR. If the request is an FIQ class interrupt, the address read from the interrupt vector table, is written to the FIQVECREG register. If the request is an IRQ class interrupt, the address is written to the IRQVECREG register and put on the VIC port of the CPU (in case of hardware vectored interrupt is enabled).

All of the interrupt registers are updated when a new high priority interrupt line becomes active.

8.4 Interrupt Vector Table (VIM RAM)

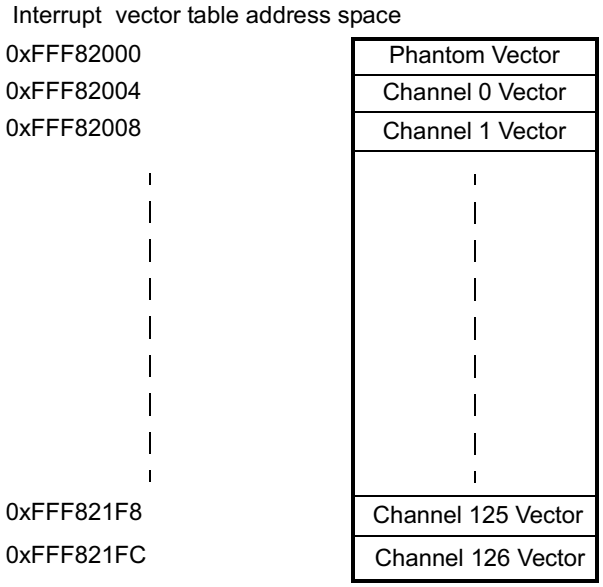
Interrupt vector table stores the address of ISRs. During register vectored interrupt and hardware vectored interrupt, VIM accesses the interrupt vector table using the vector value to fetch the address of the corresponding ISR.

For safety reasons, the interrupt vector table has protection by ECC to indicate corruption due to soft errors. The ECC scheme is implemented as a continuous background check based on memory access. The ECC logic inside VIM supports Single bit Error Correction and Double bit Error Detection (SECDED). Section 8.4.1 through Section 8.4.4 describe how ECC works in the interrupt vector table.

8.4.1 Interrupt Vector Table Operation

The interrupt vector table is organized in 128 words of 32 bits. Figure 8-7 shows the interrupt memory mapping. The table base address is 0xFFF82000.

Figure 8-7. VIM Interrupt Address Memory Map



NOTE: The interrupt vector table only has 128 entries, one phantom vector and 127 interrupt channels. Channel 127 does not have a dedicated vector and shall not be used.

There are seven bits of ECC per 32-bit ISR address. When a write is performed into the interrupt vector table, the ECC bits are calculated for the 32-bit word and written into the corresponding ECC region of interrupt vector table if ECC is enabled in VIM.

NOTE: Only 32-bit write/read access are allowed on interrupt vector table if ECC is required. Non 32-bit access might result in ECC errors.

When a read occurs from the CPU or VIM, the VIM calculates the ECC bits from the data coming from the interrupt vector table and compares it to the known good ECC value stored in the table. If a single bit error is detected in the data, the SECDED block will automatically correct it. The read data will be a corrected one in this case. If double bit errors are detected, the read data will be the uncorrected one. The access of the data and the ECC bits are performed in the same clock cycle.

The Double Bit Error (DBE) and Single Bit Error (SBE) events will be generated only if the ECC feature is enabled by ECCENA field. Correction of the data upon a SBE event will be done only if enabled EDAC_MODE field. Any double bit error will be flagged out to ESM module and as UERR flag in ECCSTAT register. The address of the data for which UERR is detected will also be stored as UERRADDR register.

Any single bit error will be registered into SBERR flag in ECCSTAT register and the corresponding address will be captured as SBERRADDR register. If SBE_INT_EN field on ECCCTL register is set to enable value, then it will be flagged out to ESM module.

Since the interrupt vector table may have an uncorrectable error (e.g. DBE), the FBVECADDR register will provide to the VIC port, IRQVECREG and FIQVECREG, a fall-back address to an ISR that can restore the interrupt vector table content. The FB_VECADDR register should be set before intializing the interrupt in the interrupt vector table, to avoid branching to an unpredictable location.

The normal operation is restored when the ECCSTAT is cleared by the CPU. It is recommended to restore the content of the VIM before clearing the ECCSTAT.

8.4.2 VIM ECC Syndrome

The VIM ECC is controlled by the ECCENA bits of ECCCTL register. After reset, the SECDED feature is disabled. The SECDED feature can be enabled by writing 0xA (1010b) in the ECCENA[3:0] bit field of the ECCCTL register.

The ECC generation is done according to the ECC syndrome table as shown in Table 8-1 and Table 8-2. Each ECC bit is build by generating the parity of the 'XOR'ed bits of the data word, whereas ECC bit 2 and 3 are even parity and the other bits odd parity

Table 8-1. ECC Syndrome Table

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ECC
x	x	x	x	x	x	x	x																	x	x	x	x	x	x	x	x	6
x	x	x	x	x	x	x	x									x	x	x	x	x	x	x	x									5
x	x							x	x	x	x	x	x			x	x							x	x	x	x	x	x			4
		x	x	x				x	x	x				x	x			x	x	x				x	x	x				x	x	3
x		x			x	x		x			x	x			x	x		x			x	x		x			x	x			x	2
			x		x		x		x		x		x	x	x				x		x		x		x		x		x	x	x	1
x		x	x		x					x		x	x	x			x			x		x	x	x	x		x				x	0

Table 8-2. ECC Error Bits for Syndrome Decode

6	5	4	3	2	1	0	ECC
x							6
	x						5
		x					4
			x				3
				x			2
					x		1
						x	0

8.4.3 Interrupt Vector Table Initialization

After reset, the interrupt vector table content, including the ECC bits is not initialized. Therefore, the interrupt vector table has to be initialized first before enabling the corresponding interrupt channel. This can be done either using the hardware initialization mechanism (in Chapter Architecture Overview) or it can be done by writing known values into the interrupt vector table by software. If ECC is required, this initialization should be done after the ECC functionality is enabled. In this way, the corresponding ECC bits will be automatically updated. This initialization is only required when vectored interrupts are used, index interrupt management does not need the table to be initialized.

1. Disable FIQ and IRQ for Cortex by writing to the CPSR register.
2. Initialize VIMRAM. See the VIMRAM initialization sequence.
3. Soft Reset VIM module(MSS_RCM:SOFTRST2:VIMRST = 0xAD) (this only resets the VIM state machine and not the VIMRAM).
4. Release reset for VIM module (MSS_RCM:SOFTRST2:VIMRST = 0x0)
5. Enable FIQ and IRQ for Cortex by writing to the CPSR register.

8.4.4 Interrupt Vector Table ECC Testing

To test the ECC checking mechanism, the ECC bits allows manual insertion of faults. This option is implemented using the TEST_DIAG_EN bit in the ECCCTL register control bit. Once TEST_DIAG_EN is enabled, the ECC bits are mapped to 0xFFF82400. In this mode, the user can modify the ECC bits without changing the data bits. If ECCENA is disabled, writing to data bits does not automatically update ECC bits. The CPU reads and writes under different conditions are summarized in Table 8-3 and Table 8-4. After that, user can force faults into either the data or the ECC bits. Finally, the ECC error can be triggered by reading interrupt vector table (not ECC bits) from VIM or CPU. Please note that no ECC checking will be done for reads of ECC bits in test mode.

Table 8-3. CPU Reads - Address Bit 10 Selects Between Normal Data and ECC Bits

VBUSP_ ADDR(10)	TEST_DIAG_ EN	ECCENA	Action
0	x(don't care)	x(don't care)	Normal RAM location read
1	x	x	ECC bits read

Table 8-4. CPU Writes - Address Bit 10 Selects Between Normal Data and ECC Bits

VBUSP_ ADDR(10)	TEST_DIAG_ EN	ECCENA	Action
0	x	1	Normal RAM locations write with ECC bits
1	0	1	This write will be blocked
1	1	1	ECC bits write
0	x	0	Normal RAM locations write without ECC bits
1	0	0	This write will be blocked
1	1	0	This write is not allowed

Following sequence should be used for injecting faults to ECC bits and testing the ECC check feature.

1. Write the data locations of VIM RAM with the required patterns while keeping ECCENA active. The ECC bits will be automatically initialized along with data bits.
2. Enable ECC test mode using TEST_DIAG_EN field of ECCCTRL register.
3. In this mode, it is possible to corrupt ECC bits using any of the following methods.
   - Read the ECC bits, flip one bit and write back
   - Read the ECC bits, flip 2 bits and write back
4. Depending on the kind corruption created, read back the data bits and check for the correction error (single bit error or double bit error or no error).

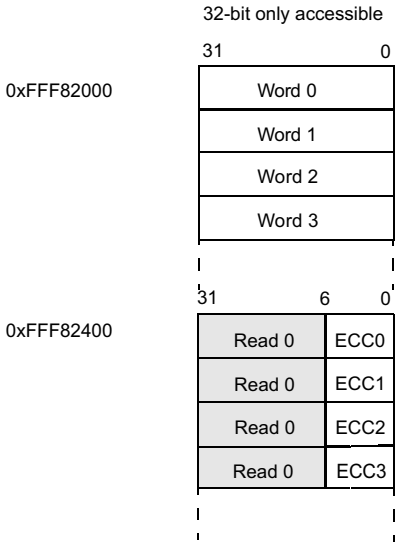
5. Read the UERRADDR and SBERRADDR registers and check for the correct address capture as well.

Following sequence should be used for injecting faults to data bits and testing the ECC check feature.

1. Write the data locations of VIM RAM with the required patterns while keeping ECCENA active. The ECC bits will be automatically initialized along with data bits.
2. Disable ECC by setting ECCENA=0 in ECCCTRL register. In this mode, writing to data bits does not automatically update ECC bits.
3. In this mode, it is possible to corrupt data bits using any of the following methods.
   - Read the data bits, flip one bit and write back
   - Read the data bits, flip 2 bits and write back
4. Depending on the kind corruption created, read back the data bits and check for the correction error (single bit error or double bit error or no error).
5. Read the UERRADDR and SBERRADDR registers and check for the correct address capture as well.

NOTE: After completing the tests for ECC check features, it should be ensured that VIM Interrupt Vector Table is initialized with valid data and corresponding check bits. Care should also be taken to clear the UERR and SBERR flag registers and the error address registers.

Figure 8-8. ECC Bits Mapping

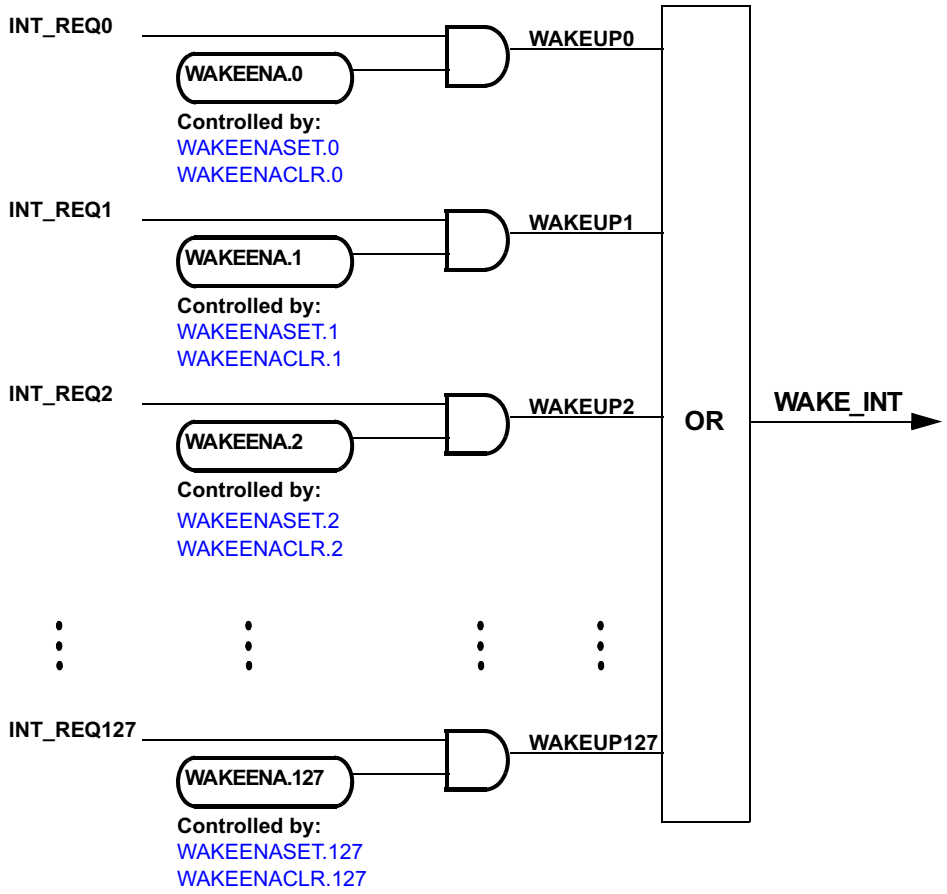


8.5 VIM Wakeup Interrupt

The wakeup interrupts are used to come out of low power mode (LPM). Any interrupt requests can be used to wake up the device. After reset, all interrupt requests are set to wake up from LPM. However, the VIM can mask unwanted interrupt lines for wake-up by using the WAKEENASET and WAKEENACLR register. The value in REQENASET / REQENACLR does NOT impact the wakeup interrupt.

As shown in Figure 8-9, the WAKEENASET and WAKEENACLR registers will enable/disable an interrupt for wake-up from low-power mode. All wake-up interrupts are “ORed” into a single signal WAKE_INT connected to the Global Clock Module.

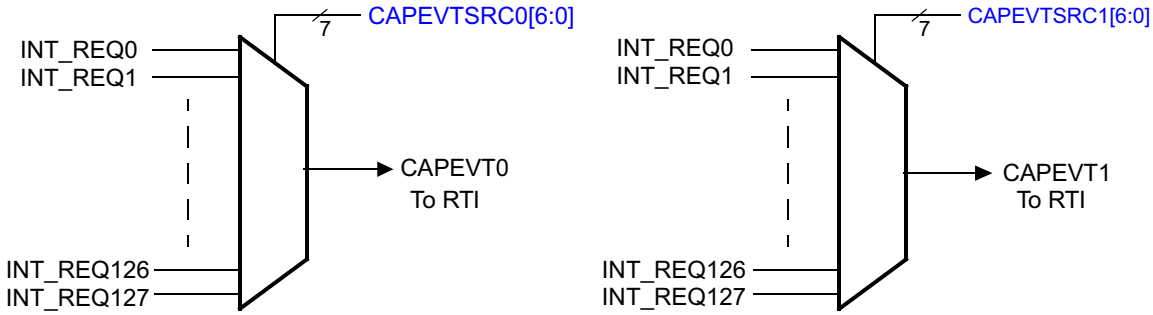
Figure 8-9. Detail of the IRQ Input



8.6 Capture Event Sources

The VIM can select any of the 128 interrupt request to generate up to two capture events for the real-time interrupt (RTI) module (see Figure 8-10). The value in REQENASET / REQENACLR does NOT impact the capture event. Two registers (Section 8.8.17) are available, one for each capture event source.

Figure 8-10. Capture Event Sources



8.7 Examples

The following sections provide examples about the operation of the VIM.

8.7.1 Examples - Configure CPU To Receive Interrupts

Example 8-1 shows how to set the vector enable (VE) bit in the CP15 R1 register to enable the hardware vector interrupt. Example 8-2 shows how to enable/disable the IRQ and FIQ through CPSR. As a convention, the program who calls these subroutines shall preserve register R1 if needed. Example 8-2 can ONLY run in privileged mode. However, in USER mode, the application software can force the program into software interrupt by instruction 'SWI'. Then, in the software interrupt service routine, user can write register SPSR, which is the copy of CPSR in this exception mode.

Example 8-1. Enable Hardware Vector Interrupt (IRQ Only)

```
_HW_Vec_Init
    MRC p15 ,#0 ,R1 ,c1 ,c0 ,#0
    ORR R1 ,R1 ,#0x01000000       ; Mask 0-31 bits except bit 24 in Sys
                                  ; Ctrl Reg of CORTEX-R4
    MCR p15 ,#0 ,R1 ,c1 ,c0 ,#0   ; Enable bit 24
    MOV PC, LR
```

Example 8-2. Enable/Disable IRQ/FIQ through CPSR

```
FIQENABLE .equ 0x40
IRQENABLE .equ 0x80
......
_Enable_Fiq
    MRS R1, CPSR
    BIC R1, R1, #FIQENABLE
    MSR CPSR, R1
    MOV PC, LR
......
_Disable_Irq
    MRS R1, CPSR
    ORR R1, R1, #IRQENABLE
    MSR CPSR, R1
    MOV PC, LR
......
_Enable_Irq
    MRS R1, CPSR
    BIC R1, R1, #IRQENABLE
    MSR CPSR, R1
    MOV PC, LR
```

8.7.2 Examples - Register Vector Interrupt and Index Interrupt Handling

Example 8-3 illustrates the configuration for the exception vectors in Register Vector Interrupt handling. After the interrupt is received by the CPU, the CPU branches to 0x18 (IRQ) or 0x1C (FIQ). The instruction placed here should be *LDR PC, [PC,#-0x1B0]*. The pending ISR address is written into the corresponding vector register (IRQVECREG for IRQ, FIQVECREG for FIQ). The CPU reads the content of the register and branches to the ISR.

Example 8-3. Exception Vector Configuration for VIM Vector

```
          .sect ".intvecs"
00000000h b _RESET             ; RESET interrupt
00000004h b _UNDEF_INST_INT    ; UNDEFINED INSTRUCTION interrupt
00000008h b _SW_INT            ; SOFTWARE interrupt
0000000Ch b _ABORT_PREF_INT    ; ABORT (PREFETCH) interrupt
00000010h b _ABORT_DATA_INT    ; ABORT (DATA) interrupt
00000014h b #-8                ; Reserved
00000018h ldr pc,[pc,#-0x1B0]  ; IRQ interrupt
0000001Ch ldr pc,[pc,#-0x1B0]  ; FIQ interrupt
```

NOTE: Program Counter (PC) always pointers two instructions beyond the current executed instruction. In this case, PC equals to '*0x18 or 0x1C + 0x08*'. The *LDR* instruction load the memory at '*PC - 0x1B0*', which is '*0x18 or 0x1C + 0x08 - 0x1B0* = 0xFFFFFE70 or 0xFFFFFE74'. These are the address of IRQVECREG and FIQVECREG, which store the pending ISR address.

Example 8-4 shows a fast response to the FIQ interrupt in Index Interrupt and can be applied to a system that has more than one channel assigned as a FIQ. It is built in Index Interrupt compatible with TMS470R1x legacy code.

Example 8-4. How to Respond to FIQ With Short Latency

```
          .sect ".intvecs"          ; Interrupt and exception vector sector
00000000h b _RESET                  ; RESET interrupt
00000004h b _UNDEF_INST_INT         ; UNDEFINED INSTRUCTION interrupt
00000008h b _SW_INT                 ; SOFTWARE interrupt
0000000Ch b _ABORT_PREF_INT         ; ABORT (PREFETCH) interrupt
00000010h b _ABORT_DATA_INT         ; ABORT (DATA) interrupt
00000014h b #-8                     ; Reserved
00000018h b _IRQ_ENTRY_0            ; IRQ interrupt
                                    ;*********************************
                                    ; INTERRUPT PROCESSING AREA
                                    ;*********************************
0000001Ch ldrb R8, [PC,#-0x21d]     ; FIQ INTERRUPT ENTRY
                                    ; R8 used to get the FIQ index
                                    ; with address pointer to the
                                    ; first FIQ banked register
00000020h ldr PC, [PC, R8, LSL#2]   ; Branch to the indexed interrupt
                                    ; routine. The prefetch
                                    ; operation causes the PC to be 2
                                    ; words (8 bytes) ahead of the
                                    ; current instruction, so
                                    ; pointing to _INT_TABLE.
00000024h nop                       ; Required due to pipeline.
                                    ;=================================
00000028h _INT_TABLE                ; FIQ INTERRUPT DISPATCH
                                    ;=================================
0000002Ch .word _FIQ_TABLE          ; beginning of FIQ Dispatch
00000030h .word _ISR1               ; dispatch to interrupt routine 1
00000034h .word _ISR2               ; dispatch to interrupt routine 2
          .
          .
```

Another way to improve the FIQ latency is to assign only one channel to the FIQ interrupt and to map the ISR code corresponding to this channel directly starting at 0x1C.

NOTE: When the CPU is in vector-enabled mode, Example 8-3 and Example 8-4 are still valid. The difference is that the CPU will not read from the 0x18 location during IRQ interrupt, but will jump directly to the corresponding ISR routine.

8.8 VIM Control Registers

Table 8-5 lists the VIM module registers. Each register begins on a word boundary. All registers are 32-bit, 16-bit, and 8-bit accessible for read and write. Write is only possible in privilege mode. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

Table 8-5. VIM Control Registers

Offset	Acronym	Register Description	Section
		ECC-related Registers	
ECh	ECCSTAT	Interrupt Vector Table ECC Status Register	Section 8.8.1
F0h	ECCCTL	Interrupt Vector Table ECC Control Register	Section 8.8.2
F4h	UERRADDR	Uncorrectable Error Address Register	Section 8.8.3
F8h	FBVECADDR	Fallback Vector Address Register	Section 8.8.4
FCh	SBERRADDR	Single Bit Error Address Register	Section 8.8.5
		Control Registers	
00h	IRQINDEX	IRQ Index Offset Vector Register	Section 8.8.7
04h	FIQINDEX	FIQ Index Offset Vector Register	Section 8.8.8
10h	FIRQPR0	FIQ/IRQ Program Control Register 0	Section 8.8.9
14h	FIRQPR1	FIQ/IRQ Program Control Register 1	Section 8.8.9
18h	FIRQPR2	FIQ/IRQ Program Control Register 2	Section 8.8.9
1Ch	FIRQPR3	FIQ/IRQ Program Control Register 3	Section 8.8.9
20h	INTREQ0	Pending Interrupt Read Location Register 0	Section 8.8.10
24h	INTREQ1	Pending Interrupt Read Location Register 1	Section 8.8.10
28h	INTREQ2	Pending Interrupt Read Location Register 2	Section 8.8.10
2Ch	INTREQ3	Pending Interrupt Read Location Register 3	Section 8.8.10
30h	REQENASET0	Interrupt Enable Set Register 0	Section 8.8.11
34h	REQENASET1	Interrupt Enable Set Register 1	Section 8.8.11
38h	REQENASET2	Interrupt Enable Set Register 2	Section 8.8.11
3Ch	REQENASET3	Interrupt Enable Set Register 3	Section 8.8.11
40h	REQENACLR0	Interrupt Enable Clear Register 0	Section 8.8.12
44h	REQENACLR1	Interrupt Enable Clear Register 1	Section 8.8.12
48h	REQENACLR2	Interrupt Enable Clear Register 2	Section 8.8.12
4Ch	REQENACLR3	Interrupt Enable Clear Register 3	Section 8.8.12
50h	WAKEENASET0	Wake-up Enable Set Register 0	Section 8.8.13
54h	WAKEENASET1	Wake-up Enable Set Register 1	Section 8.8.13
58h	WAKEENASET2	Wake-up Enable Set Register 2	Section 8.8.13
5Ch	WAKEENASET3	Wake-up Enable Set Register 3	Section 8.8.13
60h	WAKEENACLR0	Wake-up Enable Clear Register 0	Section 8.8.14
64h	WAKEENACLR1	Wake-up Enable Clear Register 1	Section 8.8.14
68h	WAKEENACLR2	Wake-up Enable Clear Register 2	Section 8.8.14
6Ch	WAKEENACLR3	Wake-up Enable Clear Register 3	Section 8.8.14
70h	IRQVECREG	IRQ Interrupt Vector Register	Section 8.8.15
74h	FIQVECREG	FIQ Interrupt Vector Register	Section 8.8.16
78h	CAPEVT	Capture Event Register	Section 8.8.17
80h-FCh	CHANCTRL	VIM Interrupt Control Register	Section 8.8.18

8.8.1 Interrupt Vector Table ECC Status Register (ECCSTAT)

Figure 8-11 and Table 8-6 describe this register.

Figure 8-11. Interrupt Vector Table ECC Status Register (ECCSTAT) [offset = ECh]

31	16
Reserved	
R-0	

15	9	8	7	1	0
Reserved		SBERR	Reserved		UERR
R-0		R/W1CP-0	R-0		R/W1CP-0

LEGEND: R/W = Read/Write; R = Read only; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Table 8-6. Interrupt Vector Table ECC Status Register (ECCSTAT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Read returns 0. Writes have no effect.
8	SBERR		The SBERR flag indicates that a single bit error has been detected and has been corrected by the SECDED logic and the Interrupt Vector Table is being used for normal operation (not bypassed).
		0	*Read:* No ECC error has occurred. *Write:* No effect.
		1	*Read:* A single bit error has occurred and was corrected by the SECDED logic *Write:* The SBERR is cleared.
7-1	Reserved	0	Read returns 0. Writes have no effect.
0	UERR		The UERR indicates that a double bit error has been found and that the Interrupt Vector Table is bypassed. The resulting vector of any IRQ/FRQ interrupt is then the value contained in the FBVECADDR register until this bit has been cleared.
		0	*Read:* No double bit error has occurred. *Write:* No effect.
		1	*Read:* A double bit error has occurred and the Interrupt Vector Table is bypassed. *Write:* The UERR is cleared and the interrupt vector can be read from the Interrupt Vector Table.

8.8.2 Interrupt Vector Table ECC Control Register (ECCCTL)

Figure 8-12. Interrupt Vector Table ECC Control Register (ECCCTL) [offset = F0h]

31 28	27 24	23 20	19 16
Reserved	SBE_EVT_EN	Reserved	EDAC_MODE
R-0	R/WP-5h	R-0	R/WP-Ah
15 12	**11 8**	**7 4**	**3 0**
Reserved	TEST_DIAG_EN	Reserved	ECCENA
R-0	R/WP-Ah	R-0	R/WP-5h

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 8-7. Interrupt Vector Table ECC Control Register (ECCCTL) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Read returns 0. Writes have no effect.
27-24	SBE_EVT_EN		These bits control the generation of Error signal out based on Single Bit Error (SBE) indications from SECDED logic for the Interrupt Vector Table.
		5h	Disable Error Event indication upon detection of SBE on the Interrupt Vector Table.
		Ah	Enable Error Event upon detection of SBE the Interrupt Vector Table.
		All other values	Writes are ignored and the values are not updated into this field. The state of the feature remains unchanged.
23-20	Reserved	0	Read returns 0. Writes have no effect.
19-16	EDAC_MODE		These bits determine whether Single Bit Errors (SBE) detected by the SECDED block will be corrected or not.
		5h	Disable correction of SBE detected by the SECDED block
		Ah	Enable correction of SBE detected by the SECDED block
		All other values	Writes are ignored and the values are not updated into this field. The state of the feature remains unchanged.
			Note: If an SBE is selected to be not corrected (using EDAC_MODE), then an SBE event will also cause VIM RAM to be bypassed just like UERR and the module to use the FBVECADDR register as the vector address.
15-12	Reserved	0	Read returns 0. Writes have no effect.
11-8	TEST_DIAG_EN		This bit maps the ECC bits into the Interrupt Vector Table frame to make them accessible by the CPU. When enabled, the ECC bits are writable as well as readable independent of data bits.
		5h	Enable memory mapping of ECC bits for read/write operation
		All other values	Disable memory mapping of ECC bits for read/write operation
			Note: To avoid soft error to disable VIM ECC mapping, it is recommended to write Ah to disable ECC bits mapping.
7-4	Reserved	0	Read returns 0. Writes have no effect.
3-0	ECCENA		VIM ECC enable.
		5h	VIM ECC is disabled.
		All other values	VIM ECC is enabled.
			Note: To avoid soft error to disable VIM ECC checking, it is recommended to write Ah to enable ECC checking.

8.8.3 Uncorrectable Error Address Register (UERRADDR)

The uncorrectable error address register gives the address of the first uncorrectable error location.

NOTE: No computation is needed when reading the complete register to retrieve the address in the Interrupt Vector Table.

This register will never be reset by a power-on reset nor any other reset source.

Figure 8-13. Uncorrectable Error Address Register (UERRADDR) [offset = F4h]

31	16
Interrupt Vector Table offset	
R-FFF8h	

15	10	9	2	1	0
Interrupt Vector Table offset		ADDERR		Word offset	
R-x		R-x		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 8-8. Uncorrectable Error Address Register (UERRADDR) Field Descriptions

Bit	Field	Description
31-10	Interrupt Vector Table offset	Interrupt Vector Table offset. Reads are always FFF8 2xxxh; writes have no effect
9-2	ADDERR	Uncorrectable error address register. This register gives the address of the first encountered double bit error since the flag has been clear. Subsequent ECC errors will not update this register until the UERR flag has been cleared. **Note: This register is valid only when PARFLG is set (see Section 8.8.1).**
1-0	Word offset	Word offset. Reads are always 0; writes have no effect.

8.8.4 Fallback Vector Address Register (FBVECADDR)

This register provides a fall-back address to the VIM if a uncorrectable error has occurred in the Interrupt Vector Table. Figure 8-14 and Table 8-9 describe this register.

NOTE: This register will never be reset by a power-on reset nor any other reset source.

Figure 8-14. Fallback Vector Address Register (FBVECADDR) [offset = F8h]

31	0
FBVECADDR	
R/WP-x	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset; x = Indeterminate

Table 8-9. Fallback Vector Address Register (FBVECADDR) Field Descriptions

Bit	Field	Description
31-0	FBVECADDR	Fallback Vector Address Register. This register is used by the VIM if the Interrupt Vector Table has been corrupted. The contents of the IRQVECREG and FIQVECREG registers will reflect the value programmed in FBVECADDR. The value provided to the VIC port will also reflect FBVECADDR until the UERR register has been cleared. This register provides the address of the ISR that will restore the integrity of the Interrupt Vector Table.

8.8.5 Single Bit Error Address Register (SBERRADDR)

This register gives the address of the first single bit ECC error detected by the ECC logic.Figure 8-15 and Table 8-10 describe this register.

NOTE: This register will never be reset by a power-on reset nor any other reset source.

Figure 8-15. Single Bit Error Address Register (SBERRADDR) [offset = FCh]

31	0
SBERRADDR	
R/WP-x	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset; x = Indeterminate

Table 8-10. Single Bit Error Address Register (SBERRADDR) Field Descriptions

Bit	Field	Description
31-0	SBERRADDR	Single Bit Error Address Register. This register gives the address of the first single bit error detected by the SECDED logic since the SBERR flag has been clear. Subsequent single bit ECC errors will not update this register until the SBERR flag has been cleared. This register provides the Interrupt Vector Table address (offset from base address word aligned) of the ECC error location. This register is valid only when the SBERR flag is set.

8.8.6 VIM Offset Vector Registers

The VIM offset register provides the user with the numerical index value that represents the pending interrupt with the highest precedence. The register IRQINDEX holds the index to the highest priority IRQ interrupt; the register FIQINDEX holds the index to the highest priority FIQ interrupt. The index can be used to locate the interrupt routine in a dispatch table, as shown in Table 8-11.

Table 8-11. Interrupt Dispatch

IRQINDEX / FIQINDEX Register Bit Field	Highest Priority Pending Interrupt Enabled
0x00	No interrupt
0x01	Channel 0
:	:
0x7F	Channel 126
0x80	Channel 127

NOTE: Channel 127 has no dedicated interrupt vector table entry. Therefore, Channel 127 shall NOT be used in application.

The VIM offset registers are read only. They are updated continuously by the VIM. When an interrupt is serviced, the offset vectors show the index for the next highest pending interrupt or 0x0 if no interrupt is pending.

8.8.7 IRQ Index Offset Vector Register (IRQINDEX)

The IRQ offset register provides the user with the numerical index value that represents the pending IRQ interrupt with the highest priority. Figure 8-16 and Table 8-12 describe this register.

Figure 8-16. IRQ Index Offset Vector Register (IRQINDEX) [offset = 00h]

31	16
Reserved	
R-0	

15	8	7	0
Reserved		IRQINDEX	
R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 8-12. IRQ Index Offset Vector Register (IRQINDEX) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Read returns 0. Writes have no effect.
7-0	IRQINDEX	0-FFh	**IRQ index vector.** The least-significant bits represent the index of the IRQ pending interrupt with the highest precedence, as shown in Table 8-11. When no interrupts are pending, the least significant byte of IRQINDEX is 0.

8.8.8 FIQ Index Offset Vector Registers (FIQINDEX)

The FIQINDEX register provides the user with a numerical index value that represents the pending FIQ interrupt with the highest priority. Figure 8-17 and Table 8-13 describe this register.

Figure 8-17. FIQ Index Offset Vector Register (FIQINDEX) [offset = F04h]

31	16
Reserved	
R-0	

15	8	7	0
Reserved		FIQINDEX	
R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 8-13. FIQ Index Offset Vector Register (FIQINDEX) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Read returns 0. Writes have no effect.
7-0	FIQINDEX	0-FFh	**FIQ index offset vector.** The least-significant bits represent the index of the FIQ pending interrupt with the highest precedence, as shown in Table 8-11. When no interrupts are pending, the least significant byte of FIQINDEX is 0x00.

8.8.9 FIQ/IRQ Program Control Registers (FIRQPR[0:3])

The FIQ/IRQ program control registers determine whether a given interrupt request will be either FIQ or IRQ. Figure 8-18, Figure 8-19, Figure 8-20, Figure 8-21 and Table 8-14 describe these registers.

NOTE: Channel 0 and 1 are FIQ only, not impacted by this register.

Figure 8-18. FIQ/IRQ Program Control Register 0 (FIRQPR0) [offset = 10h]

31	16
FIRQPR0[31:16]	
R/WP-0	

15	2	1	0
FIRQPR0[15:2]		Reserved	
R/WP-0		R-3h	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-19. FIQ/IRQ Program Control Register 1 (FIRQPR1) [offset = F14h]

31	0
FIRQPR1[63:32]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-20. FIQ/IRQ Program Control Register 2 (FIRQPR2) [offset = 18h]

31	0
FIRQPR2[95:64]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-21. FIQ/IRQ Program Control Register 3 (FIRQPR3) [offset = 1Ch]

31	0
FIRQPR3[127:96]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

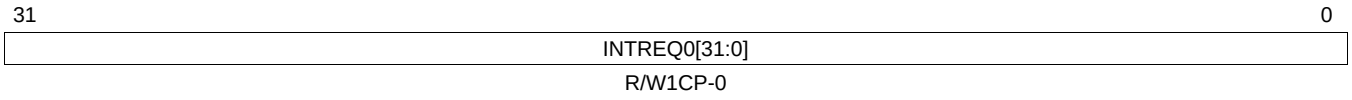
Table 8-14. FIQ/IRQ Program Control Registers (FIRQPR) Field Descriptions

Bit	Field	Value	Description
127-2	FIRQPRx[*n*]		**FIQ/IRQ program control bits.** These bits determine whether an interrupt request from a peripheral is of type FIQ or IRQ. Bit FIRQPRx[127:2] corresponds to request channel[127:2].
		0	Interrupt request is of IRQ type.
		1	Interrupt request is of FIQ type.
1-0	Reserved	3h	Read only. Writes have no effect.

8.8.10 Pending Interrupt Read Location Registers (INTREQ[0:3])

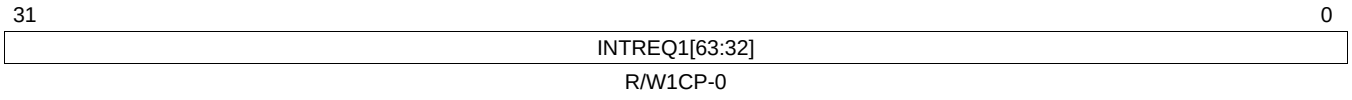
The pending interrupt register gives the pending interrupt requests. The register is updated every vbus clock cycle. Figure 8-22, Figure 8-23, Figure 8-24, Figure 8-25 and Table 8-15 describe this register.

Figure 8-22. Pending Interrupt Read Location Register 0 (INTREQ0) [offset = 20h]



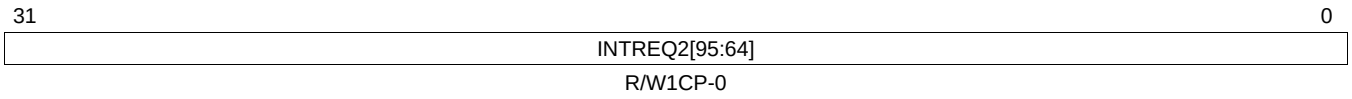
LEGEND: R/W = Read/Write; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Figure 8-23. Pending Interrupt Read Location Register 1 (INTREQ1) [offset = 24h]



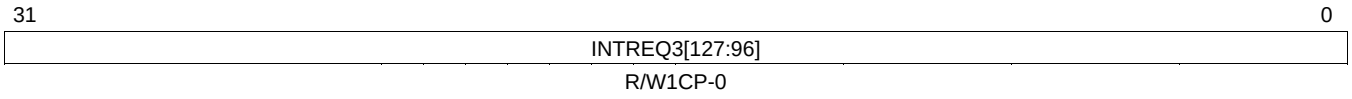
LEGEND: R/W = Read/Write; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Figure 8-24. Pending Interrupt Read Location Register 2 (INTREQ2) [offset = 28h]



LEGEND: R/W = Read/Write; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Figure 8-25. Pending Interrupt Read Location Register 3 (INTREQ3) [offset = 2Ch]



LEGEND: R/W = Read/Write; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Table 8-15. Pending Interrupt Read Location Registers (INTREQ) Field Descriptions

Bit	Field	Value	Description
127-0	INTREQx[*n*]		**Pending interrupt bits.** These bits determine whether an interrupt request is pending for the request channel between 0 and 127. The interrupt ENABLE register does not affect the value of the interrupt pending bit. Bit INTREQx[127:0] corresponds to request channel[127:0].
			User and Privilege Mode read:
		0	No interrupt event has occurred.
		1	An interrupt is pending.
			Privilege Mode write only:
		0	Writing 0 has no effect.
		1	Clears the "interrupt pending" status flag. This write-clear functionality is intended to allow clearing those interrupts which have been signaled to VIM before enabling the interrupt channel, if they are undesired.

8.8.11 Interrupt Enable Set Registers (REQENASET[0:3])

The interrupt register enable selectively enables individual request channels. Figure 8-26, Figure 8-27, Figure 8-28, Figure 8-29 and Table 8-16 describe these registers.

NOTE: Channel 0 and 1 are always enabled, not impacted by this register.

Figure 8-26. Interrupt Enable Set Register 0 (REQENASET0) [offset = 30h]

31	16
REQENASET0[31:16]	
R/WP-0	

15	2	1	0
REQENASET0[15:2]		Reserved	
R/WP-0		R-3h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Figure 8-27. Interrupt Enable Set Register 1 (REQENASET1) [offset = 34h]

31	0
REQENASET1[63:32]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-28. Interrupt Enable Set Register 2 (REQENASET2) [offset = 38h]

31	0
REQENASET2[95:64]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-29. Interrupt Enable Set Register 3 (REQENASET3) [offset = 3Ch]

31	0
REQENASET3[127:96]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Table 8-16. Interrupt Enable Set Registers (REQENASET) Field Descriptions

Bit	Field	Value	Description
127-2	REQENASETx[*n*]		**Request enable set bits.** This vector determines whether the interrupt request channel is enabled. Bit REQENASETx[127:2] corresponds to request channel[127:2].
		0	*Read:* Interrupt request channel is disabled. *Write:* No effect.
		1	*Read or Write:* The interrupt request channel is enabled.
1-0	Reserved	3h	Read only. Writes have no effect.

8.8.12 Interrupt Enable Clear Registers (REQENACLR[0:3])

The interrupt register enable selectively disables individual request channels. Figure 8-30, Figure 8-31, Figure 8-32, Figure 8-33 and Table 8-17 describe these registers.

NOTE: Channel 0 and 1 are always enabled, not impacted by this register.

Figure 8-30. Interrupt Enable Clear Register 0 (REQENACLR0) [offset = 40h]

31	16
REQENACLR0[31:16]	
R/WP-0	

15	2	1	0
REQENACLR0[15:2]		Reserved	
R/WP-0		R-3h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Figure 8-31. Interrupt Enable Clear Register 1 (REQENACLR1) [offset = 44h]

31	0
REQENACLR1[63:32]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-32. Interrupt Enable Clear Register 2 (REQENACLR2) [offset = 48h]

31	0
REQENACLR2[95:64]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-33. Interrupt Enable Clear Register 3 (REQENACLR3) [offset = 4Ch]

31	0
REQENACLR3[127:96]	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Table 8-17. Interrupt Enable Clear Registers (REQENACLR) Field Descriptions

Bit	Field	Value	Description
127-2	REQENACLRx[*n*]		**Request enable clear bits.** This vector determines whether the interrupt request channel is enabled. Bit REQENACLRx[127:2] corresponds to request channel[127:2].
		0	*Read:* Interrupt request channel is disabled. *Write:* No effect.
		1	*Read:* The interrupt request channel is enabled. *Write:* The interrupt request channel is disabled.
1-0	Reserved	3h	Read only. Writes have no effect.

8.8.13 Wake-Up Enable Set Registers (WAKEENASET[0:3])

The wake-up enable registers selectively enables individual wake-up interrupt request lines. Figure 8-34, Figure 8-35, Figure 8-36, Figure 8-37 and Table 8-18 describe these registers.

Figure 8-34. Wake-Up Enable Set Register 0 (WAKEENASET0) [offset = 50h]

31	0
WAKEENASET0[31:0]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-35. Wake-Up Enable Set Register 1 (WAKEENASET1) [offset = 54h]

31	0
WAKEENASET1[63:32]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-36. Wake-Up Enable Set Register 2 (WAKEENASET2) [offset = 58h]

31	0
WAKEENASET2[95:64]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-37. Wake-Up Enable Set Register 3 (WAKEENASET3) [offset = 5Ch]

31	0
WAKEENASET3[127:96]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Table 8-18. Wake-Up Enable Set Registers (WAKEENASET) Field Descriptions

Bit	Field	Value	Description
127-0	WAKEENASETx[*n*]		**Wake-up enable set bits.** This vector determines whether the wake-up interrupt line is enabled. Bit WAKEENASETx[127:0] corresponds to interrupt request channel[127:0].
		0	*Read:* Interrupt request channel is disabled. *Write:* No effect.
		1	*Read or Write:*The interrupt request channel is enabled.

8.8.14 Wake-Up Enable Clear Registers (WAKEENACLR[0:3])

The wake-up enable register selectively disables individual wake-up interrupt request lines. Figure 8-38, Figure 8-39, Figure 8-40, Figure 8-41 and Table 8-19 describe these registers.

Figure 8-38. Wake-Up Enable Clear Register 0 (WAKEENACLR0) [offset = 60h]

31	0
WAKEENACLR0[31:0]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-39. Wake-Up Enable Clear Register 1 (WAKEENACLR1) [offset = 64h]

31	0
WAKEENACLR1[63:32]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-40. Wake-Up Enable Clear Register 2 (WAKEENACLR2) [offset = 68h]

31	0
WAKEENACLR2[95:64]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Figure 8-41. Wake-Up Enable Clear Register 3 (WAKEENACLR3) [offset = 6Ch]

31	0
WAKEENACLR3[127:96]	
R/WP-FFFF FFFFh	

LEGEND: R/W = Read/Write; WP = Write in privilege mode only; -*n* = value after reset

Table 8-19. Wake-Up Enable Clear Registers (WAKEENACLR) Field Descriptions

Bit	Field	Value	Description
127-0	WAKEENACLRx[*n*]		**Wake-up enable clear bits.** This vector determines whether the wake-up interrupt line is enabled. Bit WAKEENACLRx[127:0] corresponds to interrupt request channel[127:0].
		0	*Read:* Wake-up interrupt channel is disabled. *Write:* No effect.
		1	*Read:* The wake-up interrupt channel is enabled. *Write:* The wake-up interrupt channel is disabled.

8.8.15 IRQ Interrupt Vector Register (IRQVECREG)

The interrupt vector register gives the address of the enabled and active IRQ interrupt. Figure 8-42 and Table 8-20 describe these registers.

Figure 8-42. IRQ Interrupt Vector Register (IRQVECREG) [offset = 70h]

31	0
IRQVECREG	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 8-20. IRQ Interrupt Vector Register (IRQVECREG) Field Descriptions

Bit	Field	Value	Description
31-0	IRQVECREG	From Section 8.4	IRQ interrupt vector register. This vector gives the address of the ISR with the highest pending IRQ request. The CPU reads the address and branches to this location.

8.8.16 FIQ Interrupt Vector Register (FIQVECREG)

The interrupt vector register gives the address of the enabled and active FIQ interrupt. Figure 8-43 and Table 8-21 describe these registers.

Figure 8-43. IRQ Interrupt Vector Register (*FIQVECREG*) [offset = 74h]

31	0
FIQVECREG	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 8-21. FIQ Interrupt Vector Register (FIQVECREG) Field Descriptions

Bit	Field	Value	Description
31-0	FIQVECREG	From Section 8.4	FIQ interrupt vector register. This vector gives the address of the ISR with the highest pending FIQ request. The CPU reads the address and branches to this location.

8.8.17 Capture Event Register (CAPEVT)

Figure 8-44 and Table 8-22 describe this register.

Figure 8-44. Capture Event Register (CAPEVT) [offset = 78h]

31	23	22	16
Reserved		CAPEVTSRC1	
R-U		R/WP-0	

15	7	6	0
Reserved		CAPEVTSRC0	
R-U		R/WP-0	

LEGEND: R = Read only; WP = Write in privilege mode only; U = Undefined; -*n* = value after reset

Table 8-22. Capture Event Register (CAPEVT) Field Descriptions

Bit	Field	Value	Description
31-23	Reserved	0	Reads are indeterminate and writes have no effect.
22-16	CAPEVTSRC1		Capture event source 1 mapping control. These bits determine which interrupt request maps to the capture event source 1 of the RTI:
		0	Interrupt request 0.
		1h	Interrupt request 1.
		:	:
		7Fh	Interrupt request 127.
15-7	Reserved	0	Reads are indeterminate and writes have no effect.
6-0	CAPEVTSRC0		Capture event source 0 mapping control. These bits determine which interrupt request maps to the capture event source 0 of the RTI:
		0	Interrupt request 0.
		1h	Interrupt request 1.
		:	:
		7Fh	Interrupt request 127.

8.8.18 VIM Interrupt Control Registers (CHANCTRL[0:31])

Thirty-two interrupt control registers control the 128 interrupt channels of the VIM. Each register controls four interrupt channels: each of them is indexed from 0 to 127. Table 8-23 shows the organization of all the registers and the reset value of each. Each four fields of the register has been named with a generic index that refers to the detailed register organization. Figure 8-45 and Table 8-24 describe these registers.

Table 8-23. Interrupt Control Registers Organization

Address	Register Acronym	Register Field 31:24 CHANMAPx$_0$	Register Field 23:16 CHANMAPx$_1$	Register Field 15:8 CHANMAPx$_2$	Register Field 7:0 CHANMAPx$_3$	Reset Value
FFFF FE80h	CHANCTRL0	CHANMAP0	CHANMAP1	CHANMAP2	CHANMAP3	0001 0203h
FFFF FE84h	CHANCTRL1	CHANMAP4	CHANMAP5	CHANMAP6	CHANMAP7	0405 0607h
:	:	:	:	:	:	:
FFFF FEF8h	CHANCTRL30	CHANMAP120	CHANMAP121	CHANMAP122	CHANMAP123	7879 7A7Bh
FFFF FEFCh	CHANCTRL31	CHANMAP124	CHANMAP125	CHANMAP126	CHANMAP127	7C7D 7E7Fh

NOTE: CHANMAP0 and CHANMAP1 are not programable. CHAN0 and CHAN1 are hard wired to INT_REQ0 and INT_REQ1.

Do NOT write any value other than 0x7F to CHANMAP127. Channel 127 is reserved because no interrupt vector table entry supports this channel.

Figure 8-45. Interrupt Control Registers (CHANCTRL[0:31]) [offset = 80h-FCh]

31	30 24	23	22 16
Rsvd	CHANMAPx$_0$	Rsvd	CHANMAPx$_1$
R-U	R/WP-n	R-U	R/WP-n

15	14 8	7	6 0
Rsvd	CHANMAPx$_2$	Rsvd	CHANMAPx$_3$
R-U	R/WP-n	R-U	R/WP-n

LEGEND: R = Read only; WP = Write in privilege mode only; U = Undefined; *-n* = value after reset (see Table 8-23)

Table 8-24. Interrupt Control Registers (CHANCTRL[0:31]) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reads are indeterminate and writes have no effect.
30-24	CHANMAPx$_0$		CHANMAPx$_0$(6-0). Interrupt CHANx$_0$ mapping control. These bits determine which interrupt request the priority channel CHANx$_0$ maps to:
		0	*Read:* Interrupt request 0 maps to channel priority CHANx$_0$. *Write:* The default value of this bit after reset is given in Table 8-23 . The channel priority CHANx$_0$ is set with the interrupt request.
		1h	*Read:* Interrupt request 1 maps to channel priority CHANx$_0$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority CHANx$_0$ is set with the interrupt request.
		:	:
		7Fh	*Read:* Interrupt request 127 maps to channel priority CHANx$_0$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority CHANx$_0$ is set with the interrupt request.
23	Reserved	0	Reads are indeterminate and writes have no effect.

Table 8-24. Interrupt Control Registers (CHANCTRL[0:31]) Field Descriptions (continued)

Bit	Field	Value	Description
22-16	$CHANMAPx_1$		$CHANMAPx_1$(6-0). Interrupt $CHANx_1$ mapping control. These bits determine which interrupt request the priority channel $CHANx_1$ maps to:
		0	*Read:* Interrupt request 0 maps to channel priority $CHANx_1$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_1$ is set with the interrupt request.
		1h	*Read:* Interrupt request 1 maps to channel priority $CHANx_1$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_1$ is set with the interrupt request.
		:	:
		7Fh	*Read:* Interrupt request 127 maps to channel priority $CHANx_1$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_1$ is set with the interrupt request.
15	Reserved	0	Reads are indeterminate and writes have no effect.
14-8	$CHANMAPx_2$		$CHANMAPx_2$(6-0). Interrupt $CHANx_2$ mapping control. These bits determine which interrupt request the priority channel $CHANx_2$ maps to:
		0	*Read:* Interrupt request 0 maps to channel priority $CHANx_2$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_2$ is set with the interrupt request.
		1h	*Read:* Interrupt request 1 maps to channel priority $CHANx_2$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_2$ is set with the interrupt request.
		:	:
		7Fh	*Read:* Interrupt request 127 maps to channel priority $CHANx_2$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_2$ is set with the interrupt request.
7	Reserved	0	Reads are indeterminate and writes have no effect.
6-0	$CHANMAPx_3$		$CHANMAPx_3$(6-0). Interrupt $CHANx_3$ mapping control. These bits determine which interrupt request the priority channel $CHANx_3$ maps to:
		0	*Read:* Interrupt request 0 maps to channel priority $CHANx_3$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_3$ is set with the interrupt request.
		1h	*Read:* Interrupt request 1 maps to channel priority $CHANx_3$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_3$ is set with the interrupt request.
		:	:
		7Fh	*Read:* Interrupt request 127 maps to channel priority $CHANx_3$. *Write:* The default value of this bit after reset is given in Table 8-23. The channel priority $CHANx_3$ is set with the interrupt request.

8.9 Interrupt Request Assignments

For a complete list of interrupt request assignments, refer to Section 1.3.6 for the 14xx, Section 2.3.9 for the 16xx/18xx, and Section 3.3.9.1 for the 68xx devices.

DSP Subsystem C674x

See the device-specific Integration section to check the availability and configuration of this module.

Topic **Page**

9.1	Introduction	1544
9.2	TMS320C674x Megamodule	1545
9.3	Memory Map	1546
9.4	Advanced Event Triggering (AET)	1546
9.5	DSP Event Assignment	1547

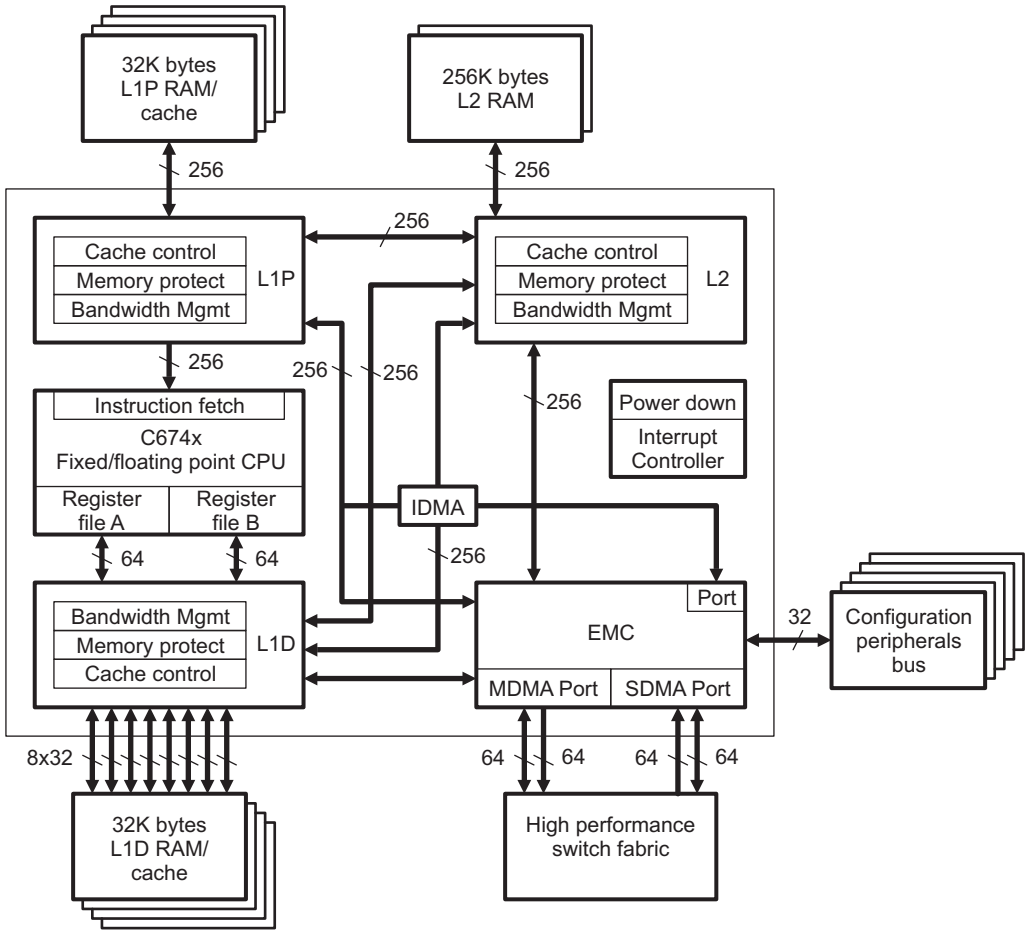
9.1 Introduction

The DSP subsystem (Figure 9-1) includes TI's standard TMS320C674x megamodule and several blocks of internal memory (L1P, L1D, and L2). This section provides an overview of the DSP subsystem and the following considerations associated with it:

- Memory mapping
- Interrupts
- Power management

For more information on the TMS320C674x megamodule, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5), the *TMS320C674x DSP CPU and Instruction Set Reference Guide* (SPRUFE8), and the *TMS320C674x DSP Cache User's Guide* (SPRUG82).

Figure 9-1. TMS320C674x Megamodule Block Diagram



9.2 TMS320C674x Megamodule

The C674x megamodule (Figure 9-1) consists of the following components:

- TMS320C674x CPU
- Internal memory controllers:
  - Level 1 program memory controller (PMC)
  - Level 1 data memory controller (DMC)
  - Level 2 unified memory controller (UMC)
  - Extended memory controller (EMC)
  - Internal direct memory access (IDMA) controller
- Internal peripherals:
  - Interrupt controller (INTC)
  - Power-down controller (PDC)
  - Bandwidth manager (BWM)
- Advanced event triggering (AET)

For more information about each of these controllers, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

9.2.1 Internal Memory Controllers

The C674x megamodule implements a two-level internal cache-based memory architecture with external memory support. Level 1 memory (L1) is split into separate program memory (L1P memory) and data memory (L1D memory). L1 memory is accessible to the CPU without stalls. Level 2 memory (L2) can also be split into L2 RAM (normal addressable on-chip memory) and L2 cache for caching external memory locations. The internal direct memory access controller (IDMA) manages DMA among the L1P, L1D, and L2 memories.

9.2.2 Internal Peripherals

The C674x megamodule includes the following internal peripherals:

- DSP interrupt controller (INTC)
- DSP power-down controller (PDC)
- Bandwidth manager (BWM)
- Internal DMA (IDMA) controller

This section briefly describes the INTC, PDC, BWM, and IDMA controller. For more information on these internal peripherals, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

9.2.2.1 Interrupt Controller (INTC)

The C674x megamodule includes an interrupt controller (INTC) to manage CPU interrupts. The INTC maps DSP device events to 12 CPU interrupts. All DSP device events are listed in . The INTC is fully described in the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

9.2.2.1.1 Interrupt Controller Registers

For more information on the DSP interrupt controller (INTC) registers, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

9.2.2.2 Power-Down Controller (PDC)

DSP core supports power-down feature. For details on how to power up/down the DSP core, see Section 4.4

9.2.2.3 Bandwidth Manager (BWM)

The bandwidth manager (BWM) provides a programmable interface for optimizing bandwidth among the requesters for resources, which include the following:

- EDMA3-initiated DMA transfers (and resulting coherency operations)
- DSP subsystem IDMA-initiated transfers (and resulting coherency operations)
- Programmable cache coherency operations
  - Block based coherency operations
  - Global coherency operations
- CPU direct-initiated transfers
  - Data access (load/store)
  - Program access

The resources include the following:

- L1P memory
- L1D memory
- L2 memory
- Resources outside of the C674x megamodule: external memory, on-chip peripherals, registers

Because any given requestor could potentially block a resource for extended periods of time, the bandwidth manager is implemented to assure fairness for all requesters.

The bandwidth manager implements a weighted-priority-driven bandwidth allocation. Each requestor (EDMA3, DSP subsystem IDMA, CPU, and so forth) is assigned a priority level on a per-transfer basis. The programmable priority level has a single meaning throughout the system. There are a total of nine priority levels, where priority zero is the highest priority and priority eight is the lowest priority. When requests for a single resource contend, access is granted to the highest-priority requestor. When the contention occurs for multiple successive cycles, a contention counter assures that the lower-priority requestor gets access to the resource every 1 out of *n* arbitration cycles, where *n* is programmable. A priority level of -1 represents a transfer whose priority has been increased due to expiration of the contention counter or a transfer that is fixed as the highest-priority transfer to a given resource.

9.2.2.4 Internal DMA (IDMA) Controller

The IDMA controller performs fast block transfers between any two memory locations local to the C674x megamodule. Local memory locations are defined as those in Level 1 program (L1P), Level 1 data (L1D), and Level 2 (L2) memories, or in the external peripheral configuration (CFG) port. The IDMA cannot transfer data to or from the internal DSP memory-mapped register space. The IDMA is fully described in the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

9.3 Memory Map

Refer to the device-specific memory map section for the addresses of the memory-map registers.

9.4 Advanced Event Triggering (AET)

The C674x megamodule supports advanced event triggering (AET). This capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- Hardware program breakpoints: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- Data watchpoints: specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- Counters: count the occurrence of an event or cycles for performance monitoring.
- State sequencing: allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

9.5 DSP Event Assignment

For DSP event assignments, refer to Table 3-9.

DSS_L3 Memory Organization

Topic **Page**

10.1 DSS_L3 Memory Organization for 14xx 1549
10.2 DSS_L3 Memory Organization for 16xx/18xx 1551
10.3 DSS_L3 Memory Organization for 68xx 1552

10.1 DSS_L3 Memory Organization for 14xx

On-chip shared memory of the 14xx device serves two main purposes:

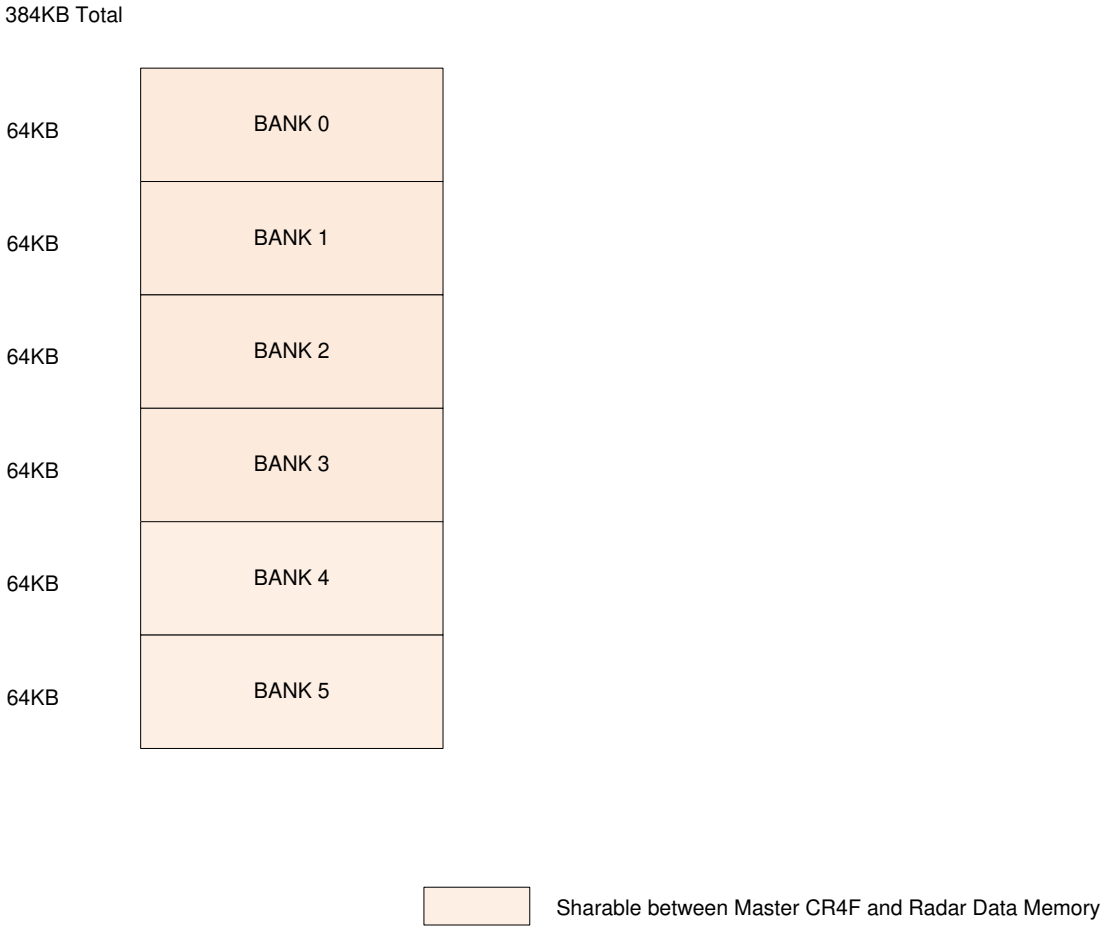
1. Radar data memory
2. A way to extend the tightly coupled memories of the Master Cortex-R4

All registers mentioned in this chapter reside in the MSS_TOPRCM module, in the device-specific control registers.

10.1.1 Functional Description

A max of 384KB of memory is available as shared memory. This memory is structured into sections of 64KB banks, as shown in Figure 10-1.

Figure 10-1. DSS_L3 Memory Organization



NOTE: Bank 0, Bank 1, Bank 2 - MSS_TCMA,DSS

Bank 3-MSS_TCMB,DSS

Bank 4, 5-DSS

Only Banks 0,1,2 has ECC

Each bank can be allotted either to MSS CR4F TCMA, MSS CR4F TCMB, or radar data memory by using register configurations to MSS_TOPRCM registers as shown in Table 10-1. Memory banks assigned to the radar data cube are accessible as DSS_L3 shared memory.

Set the appropriate bit in range bit[5:0] in the following register to assign the available bank to the desired master.

Table 10-1. MSS_TOPRCM Registers

MSS_TOPRCM Register	Description	Applicable devices
DSSMEMBANKEN	Setting bit N=1 in this register will allow shared memory's Nth bank to be accessed as Radar Data Memory. (L3 shared memory). Note: Only bits [5:0] are valid for 14xx	14xx
TCMAMEMBANK_EN	Setting bit N=1 in this register will allow shared memory's Nth bank to be accessed as TCMA ram by Master Subsystem Cortex R4F Note: Only bits [5:0] are valid for 14xx	14xx
TCMBMEMBANKEN	Setting bit N=1 in this register will allow shared memory's Nth bank to be accessed as TCMB ram by Master Subsystem Cortex R4F Note: Only bits [5:0] are valid for 14xx	14xx

NOTE: The user must ensure that each bank is assigned to only one master while writing to the above registers.

Based on the number of bits set as 1 in the xMEMBANKEN registers, that many multiples of 64KB memory spaces are allotted to the desired master. Each of these masters has a xMEMTAB register that decides which 64Kb address range is routed to which available shared memory bank. This can be set through the following MSS_TOPRCM registers, as shown in Table 10-2.

Table 10-2. MSS_TOPRCM Registers

MSS_TOPRCM Register	Description	Applicable devices
DSSMEMTAB0	Each consecutive 4 bits represent the shared memory number to which Nth 64KB address from Radar Data memory (L3 shared memory) get routed to. Bits [3:0] represent the first 64Kb address range in L3 shared memory.	14xx
TCMAMEMTAB0	Each consecutive 4 bits represent the shared memory number to which Nth 64KB address in TCMA range get routed to. Bits [3:0] represent the first 64Kb address range in L3 shared memory.	14xx
TCMBMEMTAB0	Each consecutive 4 bits represent the shared memory number to which Nth 64KB address TCMB range get routed to. Bits [3:0] represent the first 64Kb address range in L3 shared memory.	14xx

NOTE: For the 14xx, the value of each 4-bit field should be between 0 to 5.

NOTE: Shared memory bank numbers written to this field should be assigned to the master in the xMEMBANKEN register.

See Section 10.1.3 for more on memory tab initialization.

10.1.2 Memory Auto Initialization

Hardware auto initialization of the L3 memory banks is supported at reset, so that false ECC failures are not reported by writing to the MEMINITSTART register. Memory initialization is provided for each bank of 64KB. The memory is initialized to all zero values, and for every 32-bit word calculated, ECC is 0x0C. Memory initialization status is available in the MEMINITDONE register. The user must program the memory address range to be initialized by programming the SHMEMINITADDR and the ECC value in the SHMEMINITECC register.

10.1.3 Memory TAB Register

The set of memory banks allocated to the either of the masters can be ordered as per the needs of the application software. The DSSMEMTAB0, TCMAMEMTAB, and TCMBMEMTAB registers are programmed accordingly to achieve the memory bank ordering.

For example, if Banks 2,3, and 4 are enabled and allocated radar data memory (L3 shared) by writing the following set of registers:

DSSMEMBANKEN = 0x0000001C

the DSS mem tab for these banks are set to 2,3, and 4, respectively, as follows:

DSSMEMTAB0 = 0x000432XX

The rest of the banks are disabled. The first 64KB accesses to the shared memory space result in access to BANK2; 128KB accesses on bank-3, and so forth.

If the DSSMEMTAB0 is programmed to a value of 0x000234XX, then the first 128KB accesses to the shared memory space result in access to bank-4, 128KB accesses on bank-3, and so forth.

10.2 DSS_L3 Memory Organization for 16xx/18xx

The on-chip shared memory of the 16xx/18xx device serves two main purposes:

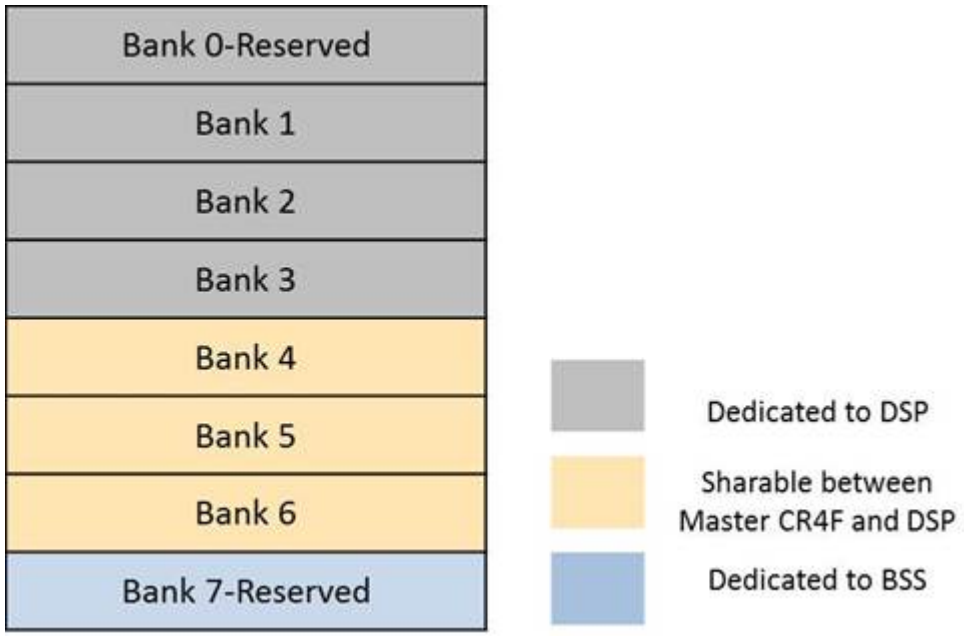
- Radar data memory
- A way to extend the tightly coupled memories of the Master Cortex-R4

All registers mentioned in this chapter reside in the MSS_TOPRCM module, in the device-specific control registers.

10.2.1 Functional Description

A maximum of 768KB of ECC-enabled memory is available as shared DSS_L3 memory in the DSP subsystem. The memory is structured into sections of 128KB, out of which can be allocated to the DSP or master Cortex-R4F, as shown in Figure 10-2.

Figure 10-2. DSS_L3 Memory Organization



As shown in Figure 10-2, the available DSS_L3 memory is 768K bytes. Some banks in the L3 shared memory are reserved (not available) or dedicated to the DSP, and the remaining memory can be shared between DSP and the Master Cortex-R4F at 128Kbyte granularity. For the Cortex-R4F, the shared memory can be an extension to the tightly coupled memories TCMA and TCMB, with the same access latencies as the TCMA and TCMB.

Any access to the DSP-dedicated space by the Master CR4F results in an ABORT. The user may decide to share the remaining memory by configuring the registers SHMEMBANKSEL. Information about how this sharing should be done is either provided by the header on the QSPI flash or over the SPI interface, before downloading the application image. ECC with the SECDED feature is supported on the DSS_L3 memory. An example of register programming for various configurations of memory allocations is shown in Table 10-3.

Table 10-3. Register Programming

DSP	CR4-TCMA	CR4-TCMB	Register Programming
768KB	256KB	192KB	SHMEMBANKSEL7TO4=0x10010101
512KB	384KB	320KB	SHMEMBANKSEL7TO4=0x10040201
512KB	512KB	192KB	SHMEMBANKSEL7TO4=0x10020201

10.2.2 Memory Auto-Initialization

Hardware auto-initialization of the DSS_L3 memory banks is supported at reset, so that false ECC failures are not reported by writing to the MSS_TOPRCM.MEMINITSTARTSHMEM register. Memory initialization is provided for each bank of 128KB. The memory is initialized to all zero values, and for every 32-bit word calculated, ECC is 0x0C. Memory initialization status is available in the MSS_TOPRCM.MEMINITDONESHMEM register.

10.2.3 Memory TAB Register

The set of memory banks allocated to the either of the master can be ordered as per the needs of application software. The DSSMEMTAB0, TCMAMEMTAB, and TCMBMEMTAB registers in MSS_TOPRCM are programmed to achieve the memory bank ordering.

For example, if banks 2, 3, and 4 are enabled or allocated for DSP by writing the following set of registers:

SHAREDBANKSEL3TO0 = 0x0101XXXX

SHAREDBANKSEL7TO4 = 0x00000001

and the DSS memory tab for these banks are set to 2, 3, and 4, respectively, as follows:

DSSMEMTAB0 = 0x000432XX

The rest of the banks are disabled by setting 0x0. Any access beyond the enabled banks wraps back to bank 2. According to this programming, the first 128KB accesses to the shared memory space result in access to BANK2, then 128KB accesses on bank-3, and so forth.

If the DSSMEMTAB0 is programmed to the value 0x000234XX, then the first 128KB accesses to the shared memory space result in access to bank-4, then 128KB accesses on bank-3, and so forth.

10.3 DSS_L3 Memory Organization for 68xx

The on-chip shared memory of the 68xx device serves two main purposes:

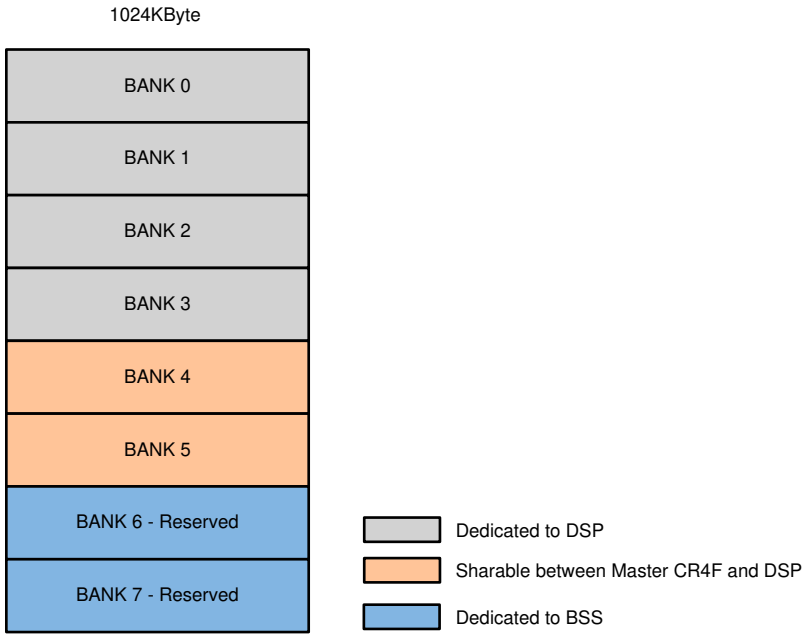
- Radar data memory
- A way to extend the tightly coupled memories of the Master Cortex-R4

All registers mentioned in this chapter reside in the MSS_TOPRCM module, in the device-specific control registers.

10.3.1 Functional Description

A maximum of 768KB of ECC-enabled memory is available as shared DSS_L3 memory in the DSP subsystem. The memory is structured into sections of 128KB, out of which can be allocated to the DSP or master Cortex-R4F, as shown in Figure 10-3.

Figure 10-3. DSS_L3 Memory Organization



As shown in Figure 10-3, the available DSS_L3 memory is 768K bytes. Some banks in the DSS_L3 memory are reserved (not available) or dedicated to the DSP, and the remaining memory can be shared between DSP and the Master Cortex-R4F at 128Kbyte granularity. For the Cortex-R4F, the shared memory can be an extension to the tightly coupled memories TCMA and TCMB, with the same access latencies as the TCMA and TCMB.

Any access to the DSP-dedicated space by the Master CR4F results in an ABORT. The user may decide to share the remaining memory by configuring the registers SHMEMBANKSEL. Information about how this sharing should be done is either provided by the header on the QSPI flash or over the SPI interface, before downloading the application image. ECC with the SECDED feature is supported on the DSS_L3 memory. An example of register programming for various configurations of memory allocations is shown in Table 10-4.

Table 10-4. Register Programming

DSP	CR4-TCMA	CR4-TCMB	Register Programming
768KB	512KB	192KB	SHMEMBANKSEL7TO4=0x10100101
512KB	640KB	320KB	SHMEMBANKSEL7TO4=0x10100402
512KB	768KB	192KB	SHMEMBANKSEL7TO4=0x10100202

10.3.2 Memory Auto-Initialization

Hardware auto-initialization of the DSS_L3 memory banks is supported at reset, so that false ECC failures are not reported by writing to the MSS_TOPRCM.MEMINITSTARTSHMEM register. Memory initialization is provided for each bank of 128KB. The memory is initialized to all zero values, and for every 32-bit word calculated, ECC is 0x0C. Memory initialization status is available in the MSS_TOPRCM.MEMINITDONESHMEM register.

10.3.3 Memory TAB Register

The set of memory banks allocated to the either of the master can be ordered as per the needs of application software. The DSSMEMTAB0, TCMAMEMTAB, and TCMBMEMTAB registers in MSS_TOPRCM are programmed to achieve the memory bank ordering.

For example, if banks 2, 3, and 4 are enabled or allocated for DSP by writing the following set of registers:

SHAREDBANKSEL3TO0 = 0x0101XXXX

SHAREDBANKSEL7TO4 = 0x00000001

and the DSS memory tab for these banks are set to 2, 3, and 4, respectively, as follows:

DSSMEMTAB0 = 0x000432XX

The rest of the banks are disabled by setting 0x0. Any access beyond the enabled banks wraps back to bank 2. According to this programming, the first 128KB accesses to the shared memory space result in access to BANK2, then 128KB accesses on bank-3, and so forth.

If the DSSMEMTAB0 is programmed to the value 0x000234XX, then the first 128KB accesses to the shared memory space result in access to bank-4, then 128KB accesses on bank-3, and so forth.

Handshake RAM (HSRAM)

The main use of these memories is to transfer the detected object list to the Master R4F from DSP. L3 shared memory can also be used for the same purpose, but considering the arbitration losses on L3, an additional memory of various size is kept for the asynchronous access by Cortex R4F without impacting the throughput, due to any kind of arbitration in the system.

Enhanced Direct Memory Access (EDMA)

This section describes the Enhanced Direct Memory Access (EDMA) controller. This chapter covers features of EDMA module. For features applicable to the EDMA instances in the device, see the device-specific Integration section. The primary purpose of the EDMA controller is to service data transfers programmed between two memory-mapped slave endpoints on the device. The EDMA controller consists of two principle blocks:

- EDMA channel controllers: EDMA_TPCC
- EDMA transfer controllers: EDMA_TPTC

Devices can have multiple instances of EDMA channel controllers, each associated with multiple EDMA transfer controllers.

The EDMA channel controller serves as the user interface for the EDMA controller. The EDMA_TPCC includes parameter RAM (PaRAM), channel control registers, and interrupt control registers. The EDMA_TPCC serves to prioritize incoming software requests or events from peripherals, and submits transfer requests (TR) to the EDMA3 transfer controller.

The EDMA transfer controllers are responsible for data movement. The transfer request packets (TRP) submitted by the EDMA_TPCC contain the transfer context, based on which the transfer controller issues read/write commands to the source and destination addresses programmed for a given transfer.

Topic **Page**

12.1 EDMA Module Overview **1557**
12.2 EDMA Controller Functional Description **1559**
12.3 EDMA Transfer Examples **1607**
12.4 EDMA Debug Checklist and Programming Tips **1614**
12.5 EDMA Request Map **1615**
12.6 EDMA Register Manual **1616**

12.1 EDMA Module Overview

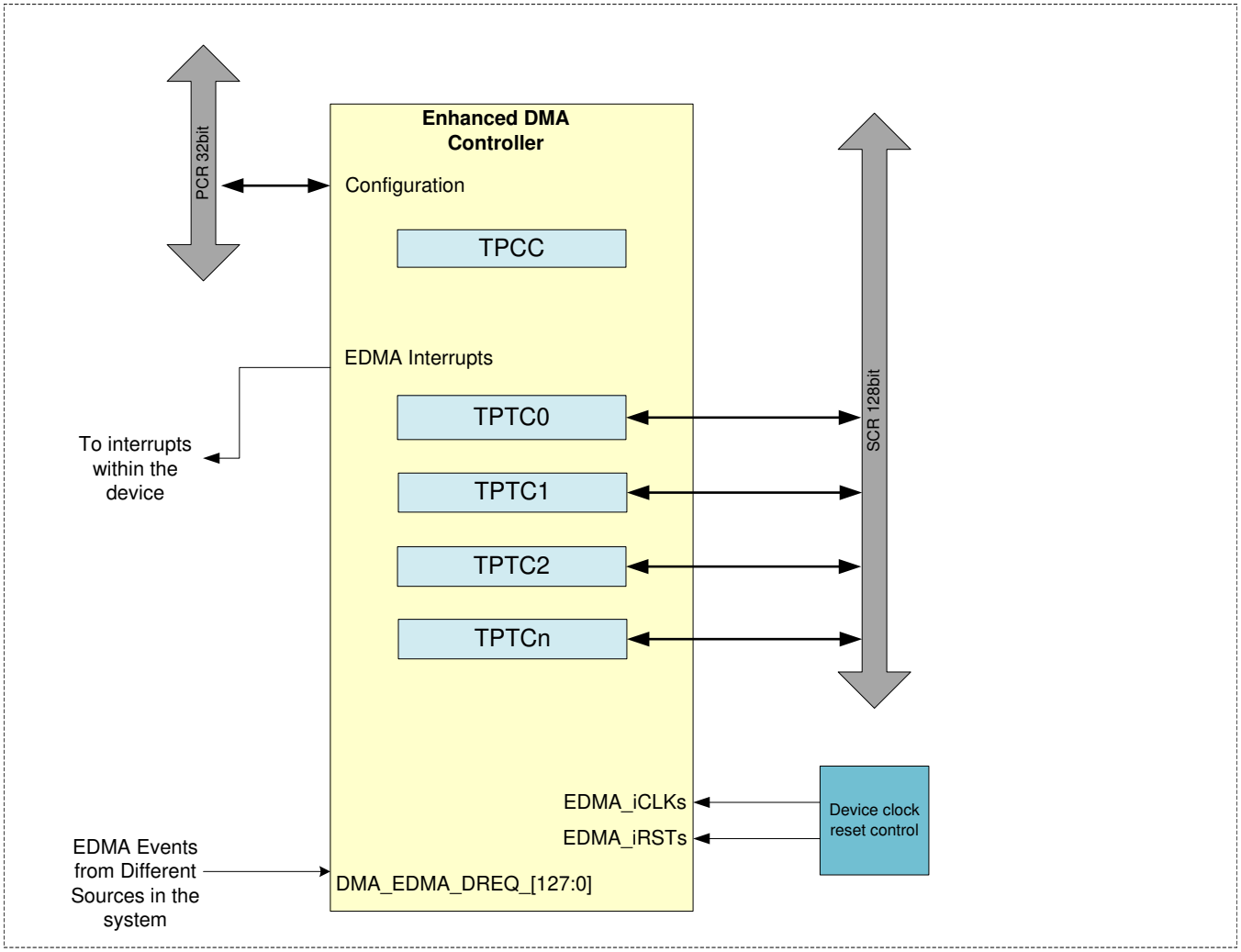
The enhanced direct memory access module, also called EDMA, performs high-performance data transfers between two slave points, memories and peripheral devices without microprocessor unit (MPU) or digital signal processor (DSP) support during transfer. EDMA transfer is programmed through a logical EDMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

The EDMA controller is based on two major principal blocks:

- EDMA third-party channel controller (EDMA_TPCC)
- EDMA third-party transfer controller (EDMA_TPTC)

Figure 12-1 shows an overview of the EDMA module.

Figure 12-1. EDMA Module Overview



For EDMA instances available on the device, see the device-specific integration section.

The **TPCC** is a high flexible channel controller that serves as both a user interface and an event interface for the EDMA controller. The EDMA_TPCC serves to prioritize incoming software requests or events from peripherals, and submits transfer requests (TRs) to the transfer controller.

The **TPTC** performs read and write transfers by EDMA ports to the slave peripherals, as programmed in the Active and Pending set of the registers. The transfer controllers are responsible for data movement, and issue read/write commands to the source and destination addresses programmed for a given transfer in the EDMA_TPCC.

12.1.1 EDMA Features

This section shows generic EDMA features. For features applicable to the EDMA instances in the devicem, see the device-specific Integration section.

The EDMA_TPCC channel controller has the following features:

- Fully orthogonal transfer description:
  - Three transfer dimensions
  - A-synchronized transfers: one dimension serviced per event
  - AB-synchronized transfers: two dimensions serviced per event
  - Independent indexes on source and destination
  - Chaining feature allowing a 3-D transfer based on a single event.
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes
  - Linking mechanism allows automatic PaRAM set update
  - Chaining allows multiple transfers to execute with one event
- Interrupt generation for the following:
  - Transfer completion
  - Error conditions
- Debug visibility:
  - Queue water marking/threshold
  - Error and status recording to facilitate debug
- 64 DMA request channels:
  - Event synchronization
  - Manual synchronization (CPUs write to event set registers EDMA_TPCC_ESR and EDMA_TPCC_ESRH).
  - Chain synchronization (completion of one transfer triggers another transfer).
- Eight QDMA channels:
  - QDMA channels trigger automatically upon writing to a parameter RAM (PaRAM) set entry.
  - Support for programmable QDMA channel to PaRAM mapping.
- 384 PaRAM sets, EDMA_TPCC0:128, EDMA_TPCC1:256:
  - Each PaRAM set can be used for a DMA channel, QDMA channel, or link set.
- Multiple transfer controllers/event queues.
- 16 event entries per event queue.

The **EDMA_TPTC** transfer controller has the following features:

- 128-bit wide read and write ports per TC
- Supports two-dimensional transfers with independent indexes on source and destination (EDMA_TPCC manages the third dimension)
- Support for increment or constant addressing mode transfers
- Interrupt and error support
- Memory-Mapped Register (MMR) bit fields are fixed position in 32-bit MMR regardless of endianness

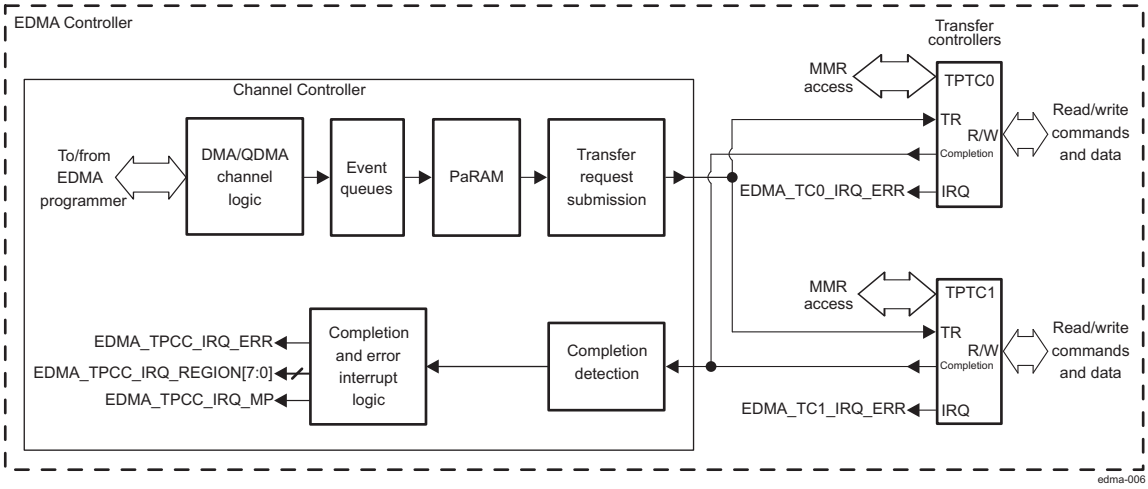
12.2 EDMA Controller Functional Description

This chapter discusses the architecture of the EDMA controller. The description contained in this section is generic to the EDMA module, and not all features mentioned here are supported by the device. See the EDMA configuration section of the device to determine the applicability of these features.

12.2.1 Block Diagram

Figure 12-2 shows the functional block diagram of the EDMA controller.

Figure 12-2. EDMA Controller Block Diagram



12.2.1.1 Third-Party Channel Controller

The TPCC is the EDMA transfer scheduler responsible for scheduling, arbitrating, and issuing user programmed transfers to the two TPTCs.

Figure 12-3 shows a functional block diagram of the EDMA channel controller (EDMA_TPCC).

The main blocks of the EDMA_TPCC are as follows:

- Parameter RAM (PaRAM): The PaRAM maintains parameter sets for channel and reload parameter sets. The PaRAM must be written with the transfer context for the desired channels and link parameter sets. EDMA_TPCC processes and sets based on a trigger event and submits a transfer request (TR) to the transfer controllers.
- EDMA event and interrupt processing registers: Allows mapping of events to parameter sets, enable/disable events, enable/disable interrupt conditions, and clearing interrupts.
- Completion detection: The completion detect block detects completion of transfers by the EDMA_TPTCs or slave peripherals. The completion of transfers can be used optionally to chain trigger new transfers or to assert interrupts.
- Event queues: Event queues form the interface between the event detection logic and the transfer request submission logic.
- Memory protection registers: Memory protection registers define the accesses (privilege level and requestor(s)) that are allowed to access the DMA channel shadow region view(s) and regions of PaRAM.

Other functions include the following:

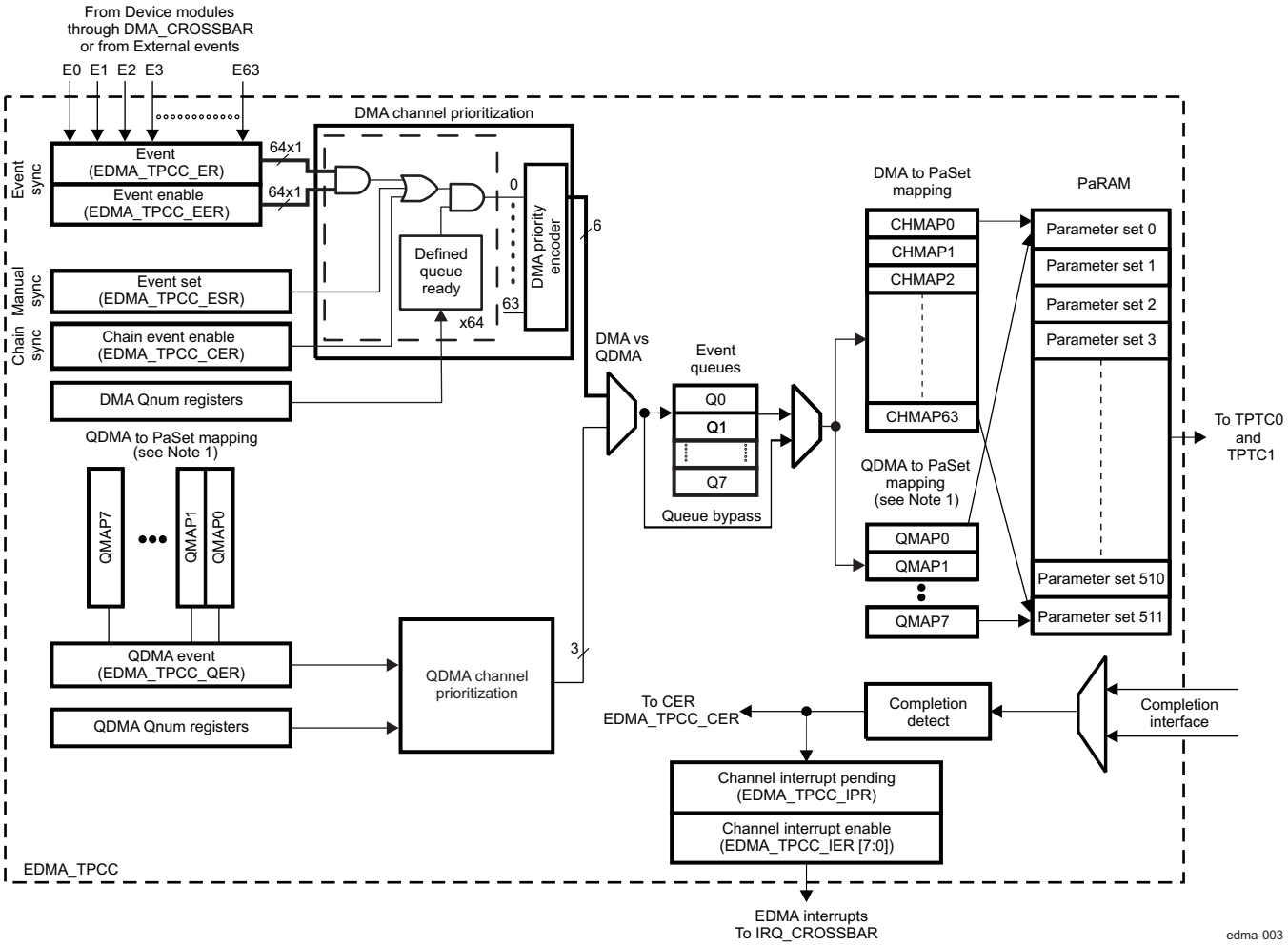
- Region registers: Region registers allow DMA resources (DMA channels and interrupts) to be assigned to unique regions that different EDMA programmers own (for example, DSPs).
- Debug registers: Debug registers allow debug visibility by providing registers to read the queue status, controller status, and missed event status.

The EDMA_TPCC includes two channel types: DMA channels (64 channels) and QDMA channels (8 channels).

Each channel is associated with a given event queue/transfer controller and with a given PaRAM set. The main difference between a DMA channel and a QDMA channel is the method that the system uses to trigger transfers.

Figure 12-3 is a block diagram of the EDMA_TPCC.

Figure 12-3. EDMA Channel Controller Block Diagram



(1) Although it is depicted twice in Figure 12-3, there is only one physical register set for the QDMA to PaRAM set mapping block.

The EDMA_TPCC supports up to 64 DMA channels and up to 8 QDMA channels. These channels are identical, except for how they are triggered:

- DMA channels are triggered by external events by the event set registers EDMA_TPCC_ESR and EDMA_TPCC_ESRH, or through chaining register EDMA_TPCC_CER.
- QDMA channels are triggered automatically (auto-triggered) by the CPU. QDMAs allow a minimum number of linear writes to be issued to the TPCC to force a series of transfers to occur.

The TPCC arbitrates among pending DMA and QDMA events with a fixed [64:1] and [8:1] priority encoder for these events, respectively (a low channel number corresponds to a high priority).

DMA events are always higher priority than QDMA events. The higher-priority event is placed in the event queue to await submission to the transfer controllers, which occurs at the earliest opportunity. Each event queue is serviced in FIFO order, with a maximum of 16 queued events per event queue. If more than one TPTC is ready to be programmed with a transmission request (TR), the event queues are serviced with fixed priority: Q0 is higher than Q1. When an event is ready to be queued and the event queue and the TC channel are empty, the event bypasses the event queue and goes directly to the PaRAM processing logic for submission to the appropriate TC. If the transfer request TR bus or PaRAM processing are busy, the bypass path is not used. The bypass is not used to dequeue for a higher-priority event.

Events are extracted from the event queue when the EDMA_TPTC is available for a new TR to be programmed into the EDMA_TPTC (signaled with the empty signal, indicating an empty program register set). As an event is extracted from the event queue, the associated PaRAM entry is processed and submitted to the TPTC as a TR. The TPCC updates the appropriate counts and addresses in the PaRAM entry in anticipation of the next trigger event for that PaRAM entry.

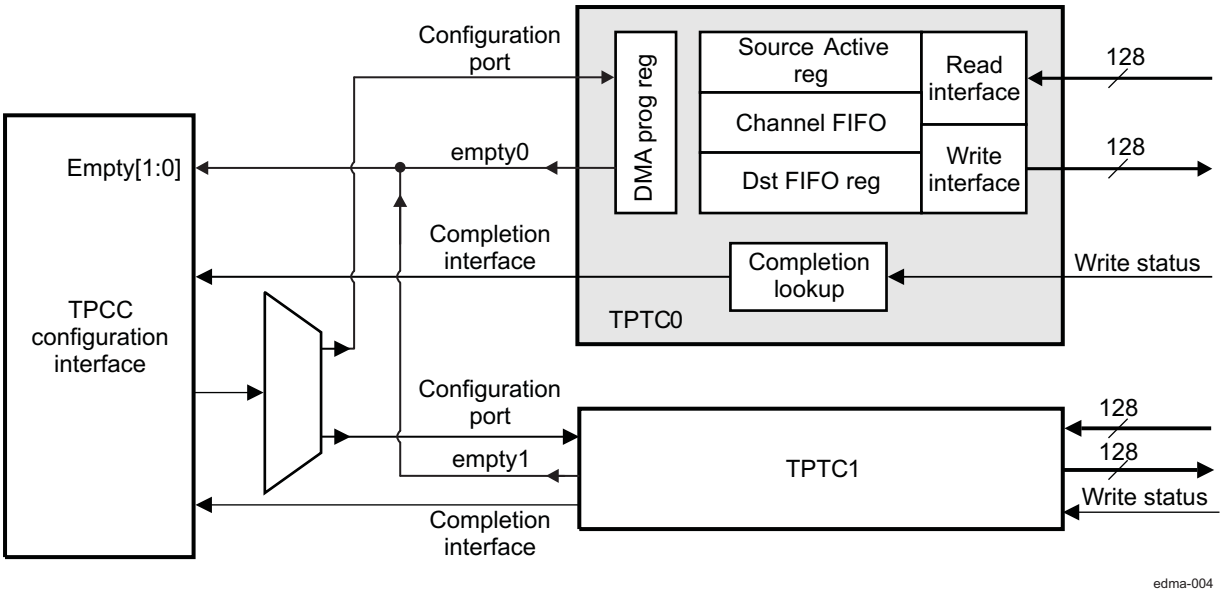
The EDMA_TPCC also has an error detection logic that causes an error interrupt generation on various error conditions (for example: missed events EDMA_TPCC_EMR and EDMA_TPCC_EMRH registers, exceeding event queue thresholds in EDMA_TPCC_CCERR register, etc.).

12.2.1.2 Third-Party Transfer Controller

The TPTC module is the EDMA transfer engine that generates transfers as programmed in dedicated working registers, using two dedicated master ports: a read-only port and a write-only port.

Figure 12-4 shows a functional block diagram and of the EDMA transfer controller (EDMA_TPTC) and its connection to the EDMA_TPCC.

Figure 12-4. TPTC Block Diagram



NOTE: The port data bus width of the instances of the TPTC is fixed at 128 bits.

Two instances of the EDMA_TPTC generate concurrent traffic on the L3_MAIN interconnect. Each TC controller consists of the following components:

- DMA Program Register Set: Stores the context for the DMA transfer that is loaded into the active register set when the current active register set completes. The CPU or TPCC programs the Program Register Set, not the active register set. For typical standalone operation, the CPU programs the Program Register while the TC services the Active register set. The Program Register set includes ownership control such that CPU software and the EDMA stay synchronized relative to one another.
- Source Active Register Set : Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in

progress in the Read Controller. The Active register set is split into independent Source and Destination, because the source interconnect controller and the distant interconnect controller operate independently of one another.

- Destination FIFO Register Set: Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in progress, or pending, in the Write Controller. The pending register must allow the source controller to begin processing a new TR while the distant register set processes the previous TR.
- Channel FIFO: Temporary holding buffer for in-flight data. The read return data of the source peripheral is stored in the Data FIFO, and then is written to the destination peripheral by the write command/data bus.
- Read Controller/Interconnect Read Interface: The Interconnect read interface issues optimally sized read commands to the source peripheral, based on a burst size of 128 bytes and available landing space in the channel FIFO.
- Write controller/Interconnect Write interface: The local interconnect write interface issues optimally sized write commands to the destination peripheral, based on a burst size of 128 bytes and available data in the channel FIFO.
- Completion interface: sends completion codes to the EDMA_TPCC when a transfer completes and generates interrupts and chained events in the TPCC module.
- Configuration port: Slave interface that provides read/write access to program registers and read access to all memory-mapped TPTC registers.

When one EDMA_TPTC module is idle and receive its first TR, DMA program register set receives the TR, where it transitions to the DMA source active set and the destination FIFO register set immediately. The second TR (if pending from EDMA_TPCC) is loaded into the DMA program set, ensuring it can start as soon as possible when the active transfer completes. As soon as the current active set is exhausted, the TR is loaded from the DMA program register set into the DMA source active register set as well as to the appropriate entry in the destination FIFO register set.

The read controller issues read commands controlled by the rules of command fragmentation and optimization. These are issued only when the data FIFO has space available for the data read. When sufficient data is in the data FIFO, the write controller starts issuing a write command again following the rules for command fragmentation and optimization.

Depending on the number of entries, the read controller can process up to two or four transfer requests ahead of the destination subject to the amount of free data FIFO.

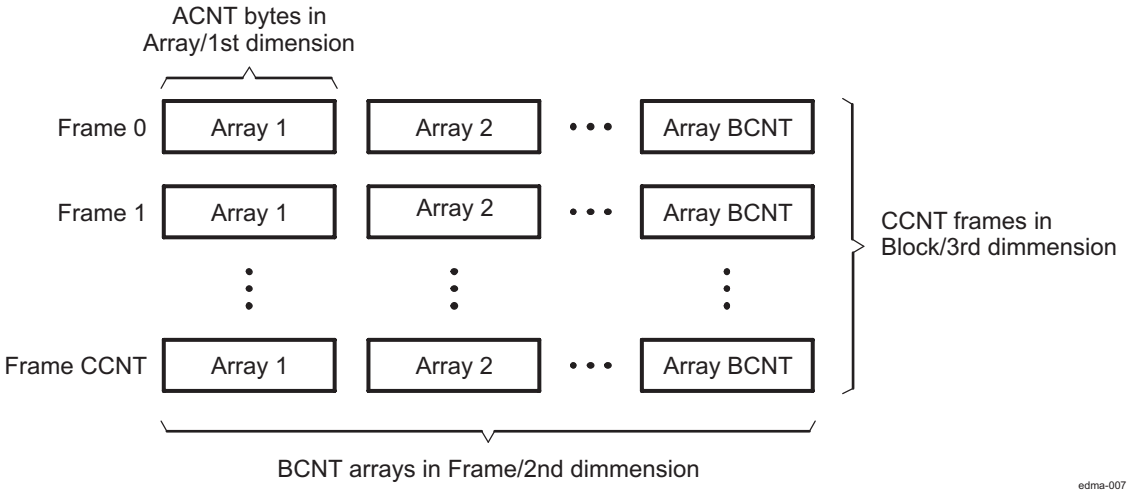
12.2.2 Types of EDMA controller Transfers

An EDMA transfer is always defined in terms of three dimensions. Figure 12-5 shows the three dimensions used by EDMA controller transfers. These three dimensions are defined as:

- 1st Dimension or Array (A): The 1st dimension in a transfer consists of EDMA_TPCC_ABCNT_n[15:0] ACNT contiguous bytes.
- 2nd Dimension or Frame (B): The 2nd dimension in a transfer consists of EDMA_TPCC_ABCNT_n[31:16] BCNT arrays of ACNT bytes. Each array transfer in the 2nd dimension is separated from each other by an index programmed using bit-fields EDMA_TPCC_BIDX_n[15:0] SBIDX or EDMA_TPCC_BIDX_n[31:16] DBIDX.
- 3rd Dimension or Block (C): The 3rd dimension in a transfer consists of CCNT frames of BCNT arrays of ACNT bytes. The Count for 3rd Dimension is defined in register EDMA_TPCC_CCNT_n[15:0] CCNT. Each transfer in the 3rd dimension is separated from the previous by an index programmed using EDMA_TPCC_CIDX_n[15:0] SCIDX or EDMA_TPCC_CIDX_n[31:16] DCIDX.

NOTE: The reference point for the index depends on the synchronization type. The amount of data transferred upon receipt of a trigger/synchronization event is controlled by the synchronization types (EDMA_TPCC_OPT_n[2] SYNCDIM bit). For these three dimensions, only two synchronization types are supported: A-synchronized transfers and AB-synchronized transfers.

Figure 12-5. Definition of ACNT, BCNT, and CCNT



12.2.2.1 A-Synchronized Transfers

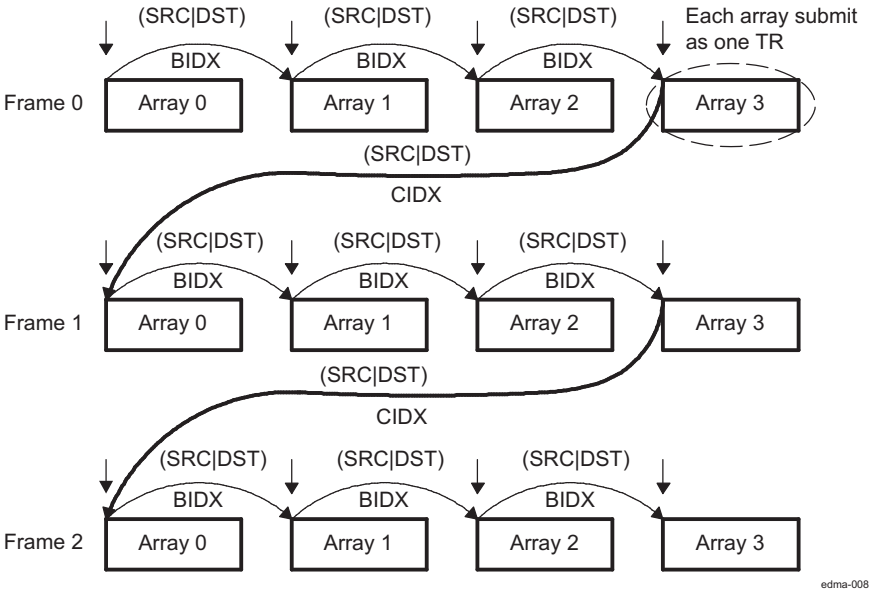
In an A-synchronized transfer, each EDMA sync event initiates the transfer of the 1st dimension of EDMA_TPCC_ABCNT_n[15:0] ACNT bytes, or one array of ACNT bytes. Each event/TR packet conveys the transfer information for one array only. Thus, BCNT × CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA_TPCC_BIDX_n[15:0]SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX, as shown in Figure 12-6, where the start address of Array N is equal to the start address of Array N – 1 plus source (SRC) or destination (DST) in EDMA_TPCC_BIDX_n register.

Frames are always separated by EDMA_TPCC_CIDX_n[15:0] SCIDX and EDMA_TPCC_CIDX_n[31:16] DCIDX. For A-synchronized transfers, after the frame is exhausted, the address is updated by adding SRCCIDX/DSTCIDX to the beginning address of the last array in the frame. As in Figure 12-6, SRCCIDX / DSTCIDX is the difference between the start of Frame 0 Array 3 to the start of Frame 1 Array 0.

Figure 12-6 shows an A-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of n (ACNT) bytes. In this example, a total of 12 sync events (BCNT × CCNT) exhaust a PaRAM set. See Figure 12-6 for details on parameter set updates.

Figure 12-6. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)



12.2.2.2 AB-Synchronized Transfers

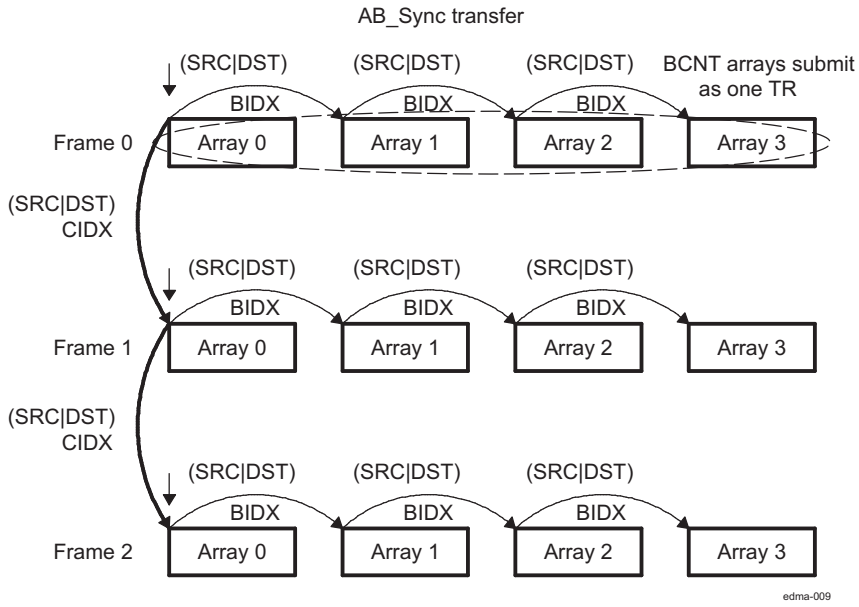
In a AB-synchronized transfer, each EDMA sync event initiates the transfer of 2 dimensions or one frame. Each event/TR packet conveys information for one entire frame of BCNT_n arrays of ACNT_n bytes. Thus, EDMA_TPCC_CCNT_n events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA_TPCC_BIDX_n[15:0] SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX as shown in Figure 12-7. Frames are always separated by SRCCIDX and DSTCIDX.

Note that for AB-synchronized transfers, after a TR for the frame is submitted, the address update is to add EDMA_TPCC_CIDX_n[15:0] SCIDX / EDMA_TPCC_CIDX_n[31:16] DCIDX to the beginning address of the beginning array in the frame. This is different from A-synchronized transfers where the address is updated by adding SRCCIDX/DSTCIDX to the start address of the last array in the frame. See Section 12.2.3.6 *Parameter Set Updates* for details on parameter set updates.

Figure 12-7 shows an AB-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of *n* (ACNT) bytes. In this example, a total of 3 sync events (CCNT) exhaust a PaRAM set; that is, a total of 3 transfers of 4 arrays each completes the transfer.

Figure 12-7. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)



NOTE: ABC-synchronized transfers are not directly supported. It can be logically achieved by chaining between multiple AB-synchronized transfers.

12.2.3 Parameter RAM (PaRAM)

The EDMA controller is a RAM-based architecture. The transfer context (source/destination addresses, count, indexes, etc.) for DMA or QDMA channels is programmed in a parameter RAM table in EDMA_TPCC. The PaRAM table is segmented into multiple PaRAM sets. Each PaRAM set includes eight four-byte PaRAM set entries (32-bytes total per PaRAM set), which includes typical DMA transfer parameters such as source address, destination address, transfer counts, indexes, options, etc.

The PaRAM structure supports flexible ping-pong, circular buffering, channel chaining, and auto-reloading (linking).

The contents of the PaRAM include the following:

- 512 PaRAM sets
- 64 channels that are direct mapped and can be used as link or QDMA sets if not used for DMA channels

- 64 channels remain for link or QDMA sets

By default, all channels map to PaRAM set to 0, they should be remapped before use by EDMA_TPCC_DCHMAPN_m and EDMA_TPCC_QCHMAPN_j registers.

Table 12-1. EDMA Parameter RAM Contents

PaRAM Set Number	Base Address	Parameters[1]
0	EDMA Base Address + 4000h to EDMA Base Address + 401Fh	PaRAM set 0
1	EDMA Base Address + 4020h to EDMA Base Address + 403Fh	PaRAM set 1
2	EDMA Base Address + 4040h to EDMA Base Address + 405Fh	PaRAM set 2
3	EDMA Base Address + 4060h to EDMA Base Address + 407Fh	PaRAM set 3
4	EDMA Base Address + 4080h to EDMA Base Address + 409Fh	PaRAM set 4
5	EDMA Base Address + 40A0h to EDMA Base Address + 40BFh	PaRAM set 5
6	EDMA Base Address + 40C0h to EDMA Base Address + 40DFh	PaRAM set 6
7	EDMA Base Address + 40E0h to EDMA Base Address + 40FFh	PaRAM set 7
8	EDMA Base Address + 4100h to EDMA Base Address + 411Fh	PaRAM set 8
9	EDMA Base Address + 4120h to EDMA Base Address + 413Fh	PaRAM set 9
...	...	...
63	EDMA Base Address + 47E0h to EDMA Base Address + 47FFh	PaRAM set 63
64	EDMA Base Address + 4800h to EDMA Base Address + 481Fh	PaRAM set 64
65	EDMA Base Address + 4820h to EDMA Base Address + 483Fh	PaRAM set 65
...	...	...
510	EDMA Base Address + 7FC0h to EDMA Base Address + 7FDFh	PaRAM set 510
511	EDMA Base Address + 7FE0h to EDMA Base Address + 7FFFh	PaRAM set 511

(1) The device has 8 QDMA channels that can be mapped to any parameter set number from 0 to 511.

12.2.3.1 PaRAM

Each parameter set of PaRAM is organized into eight 32-bit words or 32 bytes, as shown in Figure 12-8 and described in Table 12-2. Each PaRAM set consists of 16-bit and 32-bit parameters.

Figure 12-8. PaRAM Set

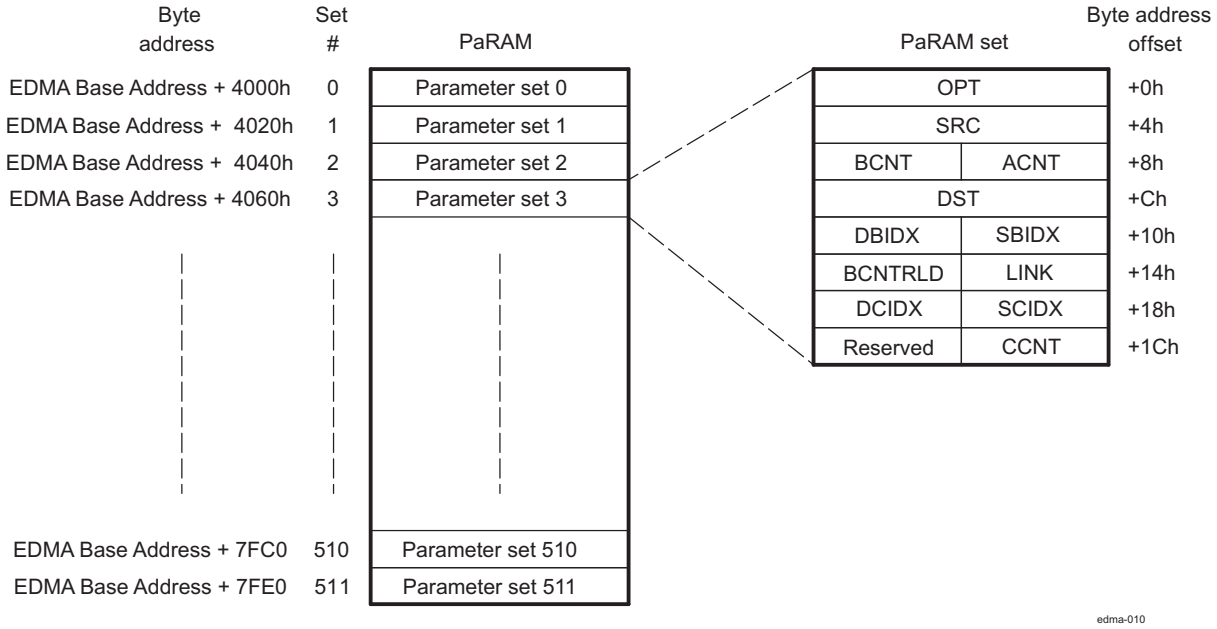


Table 12-2. EDMA Channel Parameter Description

Offset Address (bytes)	Acronym	Parameter	Description
0h	OPT	Channel Options EDMA_TPCC_OPT_n register	Transfer configuration options
4h	SRC	Channel Source Address EDMA_TPCC_SRC_n register	The byte address from which data is transferred
8h[1]	ACNT	Count for 1st Dimension EDMA_TPCC_ABCNT_n[15:0] ACNT bit-field.	Unsigned value specifying the number of contiguous bytes within an array (first dimension of the transfer). Valid values range from 1 to 65 535.
	BCNT	Count for 2nd Dimension EDMA_TPCC_ABCNT_n[31:16] BCNT bit-field.	Unsigned value specifying the number of arrays in a frame, where an array is ACNT bytes. Valid values range from 1 to 65 535.
Ch	DST	Channel Destination Address EDMA_TPCC_DST_n register	The byte address to which data is transferred
10h[1]	SBIDX	Source BCNT Index EDMA_TPCC_BIDX_n[15:0] SBIDX bit-field.	Signed value specifying the byte address offset between source arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
	DBIDX	Destination BCNT Index EDMA_TPCC_BIDX_n[31:16] DBIDX bit-field.	Signed value specifying the byte address offset between destination arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
14h[1]	LINK	Link Address EDMA_TPCC_LNK_n[15:0] LINK bit-field	The PaRAM address containing the PaRAM set to be linked (copied from) when the current PaRAM set is exhausted. A value of FFFFh specifies a null link.
	BCNTRLD	BCNT Reload EDMA_TPCC_LNK_n[31:16] BCNTRLD bit-field	The count value used to reload BCNT when BCNT decrements to 0 (TR is submitted for the last array in 2nd dimension). Only relevant in A-synchronized transfers.
18h[1]	SCIDX	Source CCNT index. EDMA_TPCC_CIDX_n[15:0] SCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last source array in a frame to the beginning of the first source array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first source array in a frame to the beginning of the first source array in the next frame.
	DCIDX	Destination CCNT index. EDMA_TPCC_CIDX_n[31:16] DCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last destination array in a frame to the beginning of the first destination array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first destination array in a frame to the beginning of the first destination array in the next frame.
1Ch	CCNT	Count for 3rd Dimension. EDMA_TPCC_CCNT_n[15:0] CCNT bit-field.	Unsigned value specifying the number of frames in a block, where a frame is BCNT arrays of ACNT bytes. Valid values range from 1 to 65 535.
	Reserved	Reserved	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

(1) If OPT, SRC, or DST is the trigger word for a QDMA transfer then it is required to do a 32-bit access to that field. Furthermore, it is recommended to perform only 32-bit accesses on the parameter RAM for best code compatibility. For example, switching the endianness of the processor will swap addresses of the 16-bit fields, but 32-bit accesses avoid the issue entirely.

12.2.3.2 EDMA Channel PaRAM Set Entry Fields

12.2.3.2.1 Channel Options Parameter (OPT)

For detailed information about the channel options parameter, see the EDMA_TPCC_OPT_n register description in , *EDMA_TPCC Register Description*.

12.2.3.2.2 Channel Source Address (SRC)

The 32-bit source address parameter specifies the starting byte address of the source. For SAM in increment mode, there are no alignment restrictions imposed by EDMA. For SAM in constant addressing mode, it must program the source address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA_TPTC returns an error. Refer to Section 12.2.12.3 *Error Generation* for additional details.

12.2.3.2.3 Channel Destination Address (DST)

The 32-bit destination address parameter specifies the starting byte address of the destination. For DAM in increment mode, there are no alignment restrictions imposed by EDMA. For DAM in constant addressing mode, it must program the destination address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA_TPTC returns an error. Refer to Section 12.2.12.3 *Error Generation* for additional details.

12.2.3.2.4 Count for 1st Dimension (ACNT)

EDMA_TPCC_ABCNT_n[15:0] ACNT represents the number of bytes within the 1st dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65 535. Therefore, the maximum number of bytes in an array is 65 535 bytes (64K – 1 bytes). ACNT must be greater than or equal to 1 for a TR to be submitted to EDMA_TPTC. A transfer with ACNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

Refer to Section 12.2.3.5 *Dummy Versus Null Transfer Comparison* and Section 12.2.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

12.2.3.2.5 Count for 2nd Dimension (BCNT)

EDMA_TPCC_ABCNT_n[15:0] BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT are between 1 and 65 535. Therefore, the maximum number of arrays in a frame is 65 535 (64K – 1 arrays). A transfer with BCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

Refer to Section 12.2.3.5 *Dummy Versus Null Transfer Comparison* and Section 12.2.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

12.2.3.2.6 Count for 3rd Dimension (CCNT)

EDMA_TPCC_CCNT_n[15:0] CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT are between 1 and 65 535. Therefore, the maximum number of frames in a block is 65 535 (64K – 1 frames). A transfer with CCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

A CCNT value of 0 is considered either a null or dummy transfer.

Refer to Section 12.2.3.5 *Dummy Versus Null Transfer Comparison* and Section 12.2.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

12.2.3.2.7 BCNT Reload (BCNTRLD)

EDMA_TPCC_LNK_n[31:16] BCNTRLD is a 16-bit unsigned value used to reload the EDMA_TPCC_ABCNT_n[15:0] BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-synchronized transfers. In this case, the EDMA_TPCC decrements the BCNT value by 1 on each TR submission. When BCNT reaches 0, the EDMA_TPCC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value.

For AB-synchronized transfers, the EDMA_TPCC submits the BCNT in the TR and the EDMA_TPTC decrements BCNT appropriately. For AB-synchronized transfers, BCNTRLD is not used.

12.2.3.2.8 Source B Index (SBIDX)

EDMA_TPCC_BIDX_n[15:0] SBIDX is a 16-bit signed value (2s complement) used for source address modification between each array in the 2nd dimension. Valid values for EDMA_TPCC_BIDX_n[15:0] SBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-synchronized and AB-synchronized transfers. Some examples:

- EDMA_TPCC_BIDX_n[15:0] SBIDX = 0000h (0): no address offset from the beginning of an array to the beginning of the next array. All arrays are fixed to the same beginning address.
- EDMA_TPCC_BIDX_n[15:0] SBIDX = 0003h (+3): the address offset from the beginning of an array to the beginning of the next array in a frame is 3 bytes. For example, if the current array begins at address 1000h, the next array begins at 1003h.
- EDMA_TPCC_BIDX_n[15:0] SBIDX = FFFFh (–1): the address offset from the beginning of an array to the beginning of the next array in a frame is –1 byte. For example, if the current array begins at address 5054h, the next array begins at 5053h.

12.2.3.2.9 Destination B Index (DBIDX)

EDMA_TPCC_BIDX_n[31:16] DBIDX is a 16-bit signed value (2s complement) used for destination address modification between each array in the 2nd dimension. Valid values for EDMA_TPCC_BIDX_n[31:16] DBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-synchronized and AB-synchronized transfers. Refer to Section 12.2.3.2.8 *Source B Index (SBIDX)* for examples.

12.2.3.2.10 Source C Index (SCIDX)

EDMA_TPCC_CIDX_n[15:0] SCIDX is a 16-bit signed value (2s complement) used for source address modification in the 3rd dimension. Valid values for EDMA_TPCC_CIDX_n[15:0] SCIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

NOTE: When SCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 12-6), while the current array in an AB-synchronized transfer is the first array in the frame (Figure 12-7).

12.2.3.2.11 Destination C Index (DCIDX)

EDMA_TPCC_CIDX_n[31:16] DCIDX is a 16-bit signed value (2s complement) used for destination address modification in the 3rd dimension. Valid values are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array TR in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

NOTE: When DCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 12-6), while the current array in a AB-synchronized transfer is the first array in the frame (Figure 12-7).

12.2.3.2.12 Link Address (LINK)

The EDMA_TPCC provides a mechanism, called linking, to reload the current PaRAM set upon its natural termination (that is, after the count fields are decremented to 0) with a new PaRAM set. The 16-bit parameter EDMA_TPCC_LNK_n[15:0] LINK specifies the byte address offset in the PaRAM from which the EDMA_TPCC loads/reloads the next PaRAM set during linking.

It must program the link address to point to a valid aligned 32-byte PaRAM set. The 5 LSBs of the LINK field should be cleared to 0.

The EDMA_TPCC ignores the upper 2 bits of the LINK entry, allowing the flexibility of programming the link address as either an absolute/literal byte address or use the PaRAM-base-relative offset address. Therefore, if it use the literal address with a range from 4000h to 7FFFh, it will be treated as a PaRAM-base-relative value of 0000h to 3FFFh.

It should check that the programed value in the EDMA_TPCC_LNK_n[15:0] LINK field is correctly, so that link update is requested from a PaRAM address that falls in the range of the available PaRAM addresses on the device.

Value of FFFFh in EDMA_TPCC_LNK_n[15:0] LINK bit-field is referred to as a NULL link that should cause the EDMA_TPCC to perform an internal write of 0 to all entries of the current PaRAM set, except for the EDMA_TPCC_LNK_n[15:0] LINK field is set to FFFFh. Also, see Section 12.2.5 *Completion of a DMA Transfer* for details on terminating a transfer.

12.2.3.3 Null PaRAM Set

A null PaRAM set is defined as a PaRAM set where all count fields (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, and EDMA_TPCC_CCNT_n[15:0] CCNT) are cleared to 0. If a PaRAM set associated with a channel is a NULL set, then when serviced by the EDMA_TPCC, the bit corresponding to the channel is set in the associated event missed register (EDMA_TPCC_EMR, EDMA_TPCC_EMRH, or EDMA_TPCC_QEMR). This bit remains set in the associated secondary event register (EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER).
This implies that any future events on the same channel are ignored by the EDMA_TPCC and it is required to clear the bit in EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER for the channel. This is considered an error condition, since events are not expected on a channel that is configured as a null transfer.

12.2.3.4 Dummy PaRAM Set

A dummy PaRAM set is defined as a PaRAM set where at least one of the count fields (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, or EDMA_TPCC_CCNT_n[15:0] CCNT) is cleared to 0 and at least one of the count fields is nonzero.

If a PaRAM set associated with a channel is a dummy set, then when serviced by the EDMA_TPCC, it will not set the bit corresponding to the channel (DMA/QDMA) in the event missed register (EDMA_TPCC_EMR, EDMA_TPCC_EMRH, or EDMA_TPCC_QEMR) and the secondary event register (EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER) bit gets cleared similar to a normal transfer. Future events on that channel are serviced. A dummy transfer is a legal transfer of 0 bytes.

12.2.3.5 Dummy Versus Null Transfer Comparison

There are some differences in the way the EDMA_TPCC logic treats a dummy versus a null transfer request. A null transfer request is an error condition, but a dummy transfer is a legal transfer of 0 bytes. A null transfer causes an error bit (E*n*) in EDMA_TPCC_EMR to get set and the E*n* bit in EDMA_TPCC_SER remains set, essentially preventing any further transfers on that channel without clearing the associated error registers.

Table 12-3 summarizes the conditions and effects of null and dummy transfer requests.

Table 12-3. Dummy and Null Transfer Request

Feature	Null TR	Dummy TR
EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR is set	Yes	No
EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER remains set	Yes	No
Link update (STATIC = 0 in EDMA_TPCC_OPT_n)	Yes	Yes
EDMA_TPCC_QER is set	Yes	Yes
EDMA_TPCC_IPR / EDMA_TPCC_IPRH, EDMA_TPCC_CER / EDMA_TPCC_CERH is set using early completion	Yes	Yes

12.2.3.6 Parameter Set Updates

When a TR is submitted for a given DMA/QDMA channel and its corresponding PaRAM set, the EDMA_TPCC is responsible for updating the PaRAM set in anticipation of the next trigger event. For events that are not final, this includes address and count updates; for final events, this includes the link update.

The specific PaRAM set entries that are updated depend on the channel's synchronization type (A-synchronized or B-synchronized) and the current state of the PaRAM set. A B-update refers to the decrementing of EDMA_TPCC_ABCNT_n[31:16] BCNT in the case of A-synchronized transfers after the submission of successive TRs. A C-update refers to the decrementing of CCNT in the case of A-synchronized transfers after BCNT TRs for EDMA_TPCC_ABCNT_n[15:0] ACNT byte transfers have submitted. For AB-synchronized transfers, a C-update refers to the decrementing of EDMA_TPCC_CCNT_n[15:0] CCNT after submission of every transfer request.

Refer to Table 12-4 for details and conditions on the parameter updates. A link update occurs when the PaRAM set is exhausted, as described in Section 12.2.3.7 *Linking Transfers*.

After the TR is read from the PaRAM (and is in process of being submitted to EDMA_TPTC), the following fields are updated if needed:

- A-synchronized: BCNT, CCNT, SRC, DST.
- AB-synchronized: CCNT, SRC, DST.

The following fields are not updated (except for during linking, where all fields are overwritten by the link PaRAM set):

- A-synchronized: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_LNK_n[31:16] BCNTRLD, EDMA_TPCC_BIDX_n[15:0] SBIDX, EDMA_TPCC_BIDX_n[31:16] DBIDX, EDMA_TPCC_CIDX_n[15:0] SCIDX, EDMA_TPCC_CIDX_n[31:16] DCIDX, EDMA_TPCC_OPT_n, EDMA_TPCC_LNK_n[15:0]LINK.
- AB-synchronized: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, EDMA_TPCC_LNK_n[31:16] BCNTRLD, EDMA_TPCC_BIDX_n[15:0] SBIDX, EDMA_TPCC_BIDX_n[31:16] DBIDX, EDMA_TPCC_CIDX_n[15:0] SCIDX, EDMA_TPCC_CIDX_n[31:16] DCIDX, EDMA_TPCC_OPT_n, EDMA_TPCC_LNK_n[15:0]LINK.

NOTE: PaRAM updates only pertain to the information that is needed to properly submit the next transfer request to the EDMA_TPTC. Updates that occur while data is moved within a transfer request are tracked within the transfer controller, and is detailed in Section 12.2.12 *EDMA Transfer Controller (EDMA_TPTC)*. For A-synchronized transfers, the EDMA_TPCC always submits a TRP for EDMA_TPCC_ABCNT_n[15:0] ACNT bytes (EDMA_TPCC_ABCNT_n[31:16] BCNT = 1 and EDMA_TPCC_CCNT_n[15:0] CCNT = 1). For AB-synchronized transfers, the EDMA_TPCC always submits a TRP for EDMA_TPCC_ABCNT_n[15:0] ACNT bytes of BCNT arrays (EDMA_TPCC_CCNT_n[15:0] CCNT = 1). The EDMA_TPTC is responsible for updating source and destination addresses within the array based on EDMA_TPCC_ABCNT_n[15:0] ACNT and EDMA_TPCC_OPT_n[10:8] FWID. For AB-synchronized transfers, the EDMA_TPTC is also responsible to update source and destination addresses between arrays based on EDMA_TPCC_BIDX_n[15:0] SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX.

Table 12-4 shows the details of parameter updates that occur within EDMA_TPCC for A-synchronized and AB-synchronized transfers.

Table 12-4. Parameter Updates in EDMA_TPCC (for Non-Null, Non-Dummy PaRAM Set)

	A-Synchronized Transfer			AB-Synchronized Transfer		
	B-Update	C-Update	Link Update	B-Update	C-Update	Link Update
Condition:	BCNT > 1	BCNT == 1 && CCNT > 1	BCNT == 1 && CCNT == 1	N/A	EDMA_TPCC_CCNT_n[15:0] CCNT > 1	EDMA_TPCC_CCNT_n[15:0] CCNT == 1
SRC	+= SBIDX	+= SCIDX	= Link.EDMA_TPCC_SRC_n	in EDMA_TPTC	+= SCIDX	= Link.EDMA_TPCC_SRC_n
DST	+= DBIDX	+= DCIDX	= Link.EDMA_TPCC_DST_n	in EDMA_TPTC	+= DCIDX	= Link.EDMA_TPCC_DST_n
ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT
BCNT	–= 1	= BCNTRLD	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT	in EDMA_TPTC	N/A	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT
CCNT	None	–= 1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT	in EDMA_TPTC	-=1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT
SBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPTC	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
SCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPTC	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0]LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0]LINK
BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD
OPT[1]	None	None	= LINK.EDMA_TPCC_OPT_n	None	None	= LINK.EDMA_TPCC_OPT_n

(1) In all cases, no updates occur if EDMA_TPCC_OPT_n[3] STATIC == 1 for the current PaRAM set.

NOTE: The EDMA_TPCC includes no special hardware to detect when an indexed address update calculation overflows/underflows. The address update will wrap across boundaries as programmed by the user. It should ensure that no transfer is allowed to cross internal port boundaries between peripherals. A single TR must target a single source/destination slave endpoint.

12.2.3.7 Linking Transfers

The EDMA_TPCC provides a mechanism known as linking, which allows the entire PaRAM set to be reloaded from a location within the PaRAM memory map (for both DMA and QDMA channels). Linking is especially useful for maintaining ping-pong buffers, circular buffering, and repetitive/continuous transfers with no CPU intervention. Upon completion of a transfer, the current transfer parameters are reloaded with the parameter set pointed that the 16-bit link address field of the current parameter set points to. Linking only occurs when the EDMA_TPCC_OPT_n[3] STATIC bit is cleared.

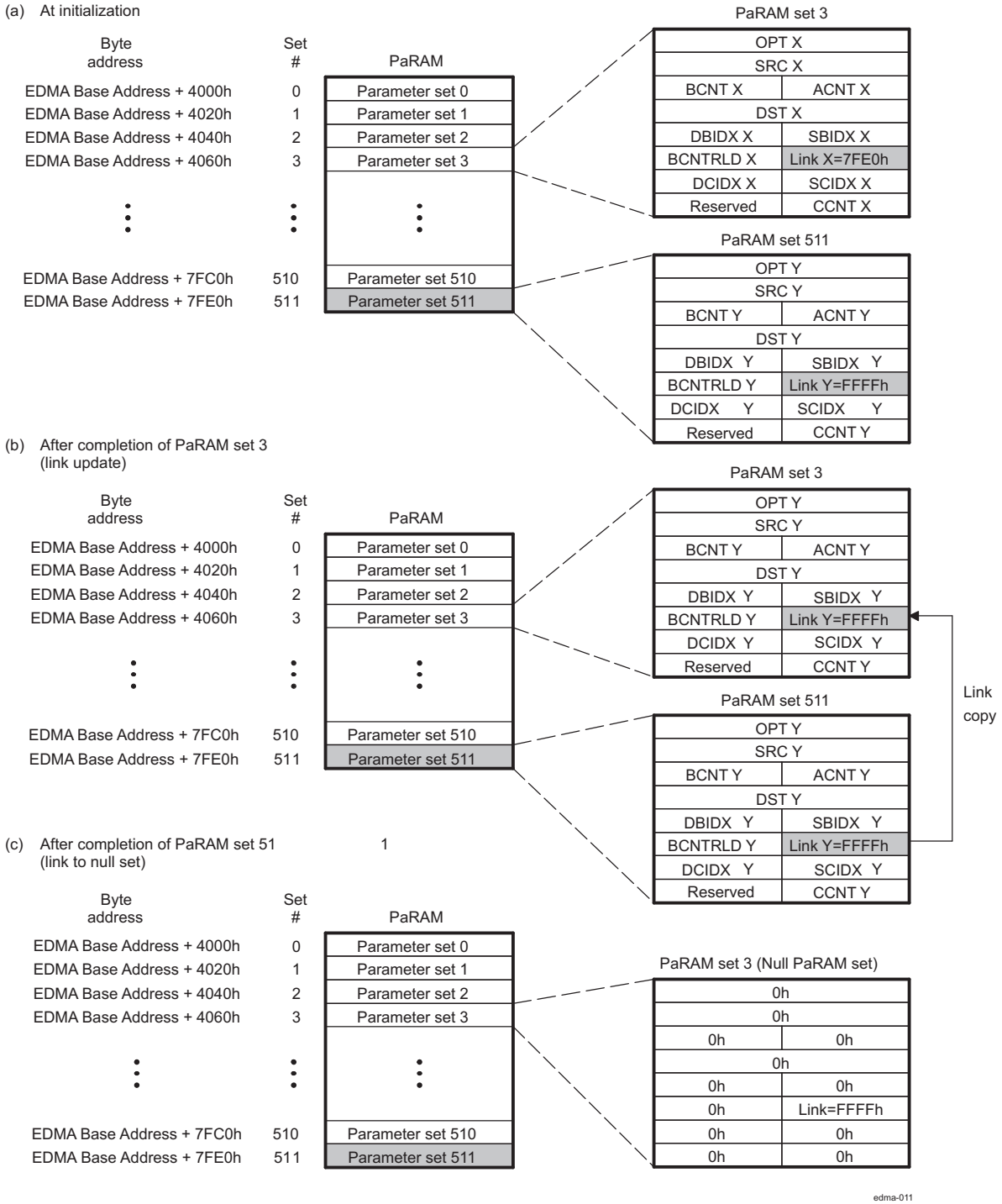
NOTE: It should always link a transfer (EDMA or QDMA) to another useful transfer. If it must terminate a transfer, then link the transfer to a NULL parameter set. Refer to Section 12.2.3.3 *Null PaRAM Set*.

The link update occurs after the current PaRAM set event parameters have been exhausted. An event's parameters are exhausted when the EDMA channel controller has submitted all of the transfers that are associated with the PaRAM set.

A link update occurs for null and dummy transfers depending on the state of the EDMA_TPCC_OPT_n[3] STATIC bit and the EDMA_TPCC_LNK_n[15:0] LINK field. In both cases (null or dummy), if the value of EDMA_TPCC_LNK_n[15:0] LINK is FFFFh, then a null PaRAM set (with all 0s and EDMA_TPCC_LNK_n[15:0] LINK set to FFFFh) is written to the current PaRAM set. Similarly, if EDMA_TPCC_LNK_n[15:0] LINK is set to a value other than FFFFh, then the appropriate PaRAM location that EDMA_TPCC_LNK_n[15:0] LINK points to is copied to the current PaRAM set.

Once the channel completion conditions are met for an event, the transfer parameters that are located at the link address are loaded into the current DMA or QDMA channel's associated parameter set. This indicates that the EDMA_TPCC reads the entire set (eight words) from the PaRAM set specified by EDMA_TPCC_LNK_n[15:0] LINK and writes all eight words to the PaRAM set that is associated with the current channel. Figure 12-9 shows an example of a linked transfer.

Figure 12-9. Linked Transfer

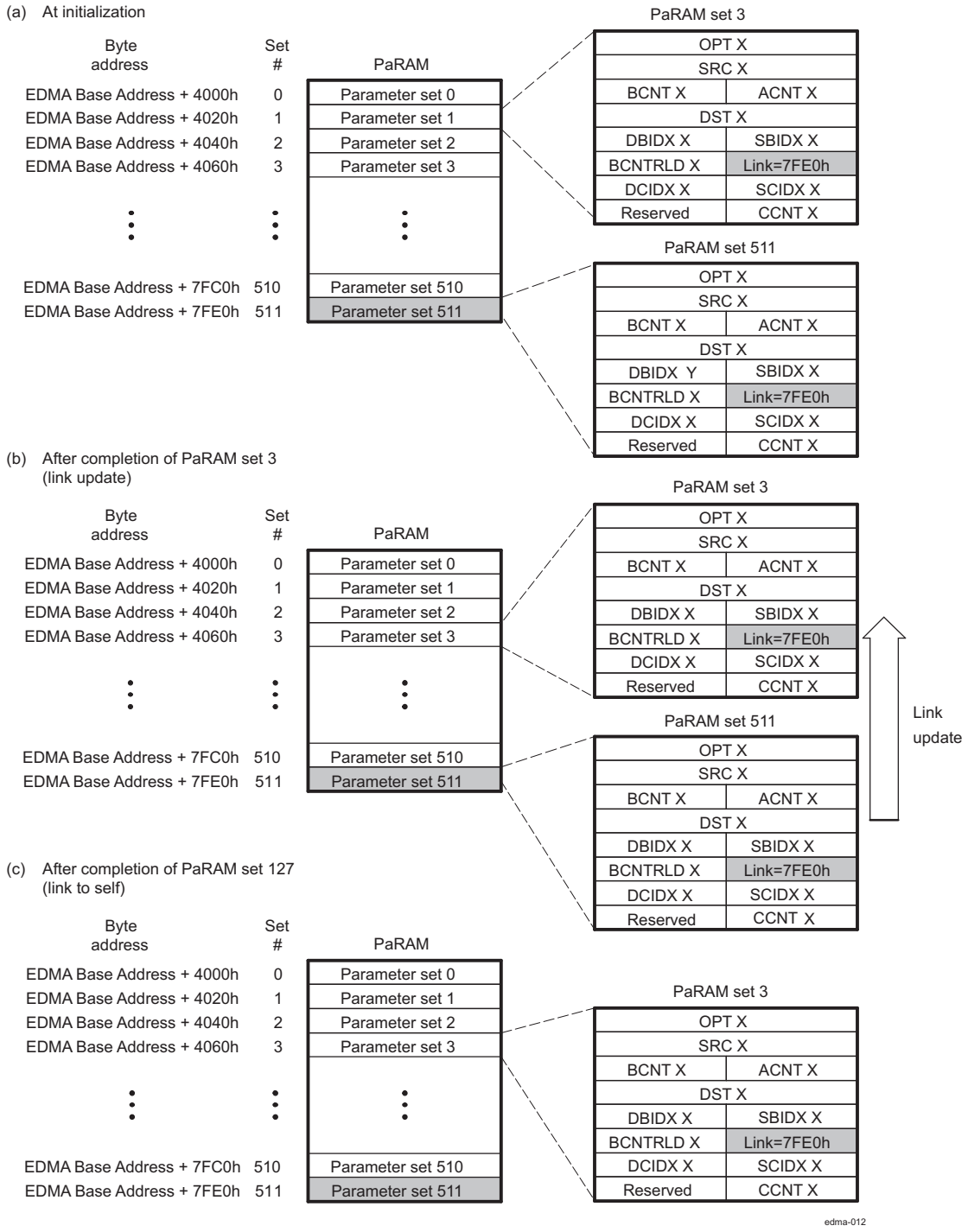


Any PaRAM set in the PaRAM can be used as a link/reload parameter set. The PaRAM sets associated with peripheral synchronization events (refer to Section 12.2.6 *Event, Channel, and PaRAM Mapping*) only use for linking if the corresponding events are disabled.

If a PaRAM set location is defined as a QDMA channel PaRAM set (by EDMA_TPCC_QCHMAPN_j register), then copying the link PaRAM set into the current QDMA channel PaRAM set is recognized as a trigger event. It is latched in EDMA_TPCC_QER because a write to the trigger word was performed. This feature is used to create a linked list of transfers using a single QDMA channel and multiple PaRAM sets. Refer to Section 12.2.4.2 *QDMA Channels*.

Linking to itself replicates the behavior of auto-initialization, thus facilitating the use of circular buffering and repetitive transfers. After an EDMA channel exhausts its current PaRAM set, it reloads all of the parameter set entries from another PaRAM set, which is initialized with values that are identical to the original PaRAM set. Figure 12-10 shows an example of a linked to self transfer. Here, the PaRAM set 511 has the link field pointing to the address of parameter set 511 (linked to self).

Figure 12-10. Link-to-Self Transfer



NOTE: If the in EDMA_TPCC_OPT_n[3] STATIC bit is set for a PaRAM set, then link updates are not performed.

12.2.3.8 Constant Addressing Mode Transfers/Alignment Issues

If either EDMA_TPCC_OPT_n[0] SAM or EDMA_TPCC_OPT_n[1] DAM is set (constant addressing mode), then the source or destination address must be aligned to a 256-bit aligned address, respectively, and the corresponding EDMA_TPCC_BIDX_n is an even multiple of 32 bytes (256 bits). The EDMA_TPCC does not recognize errors here, but the EDMA_TPTC asserts an error if this is not true. Refer to Section 12.2.12.3 *Error Generation*.

NOTE: The constant addressing (CONST) mode has limited applicability. The EDMA is configured for the constant addressing mode (EDMA_TPCC_OPT_n[0] SAM / EDMA_TPCC_OPT_n[1] DAM = 1) only if the transfer source or destination (on-chip memory, off-chip memory controllers, slave peripherals) support the constant addressing mode. If the constant addressing mode is not supported, the similar logical transfer can be achieved using the increment (INCR) mode (EDMA_TPCC_OPT_n[0] SAM / EDMA_TPCC_OPT_n[1] DAM =0) by appropriately programming the count and indices values.

12.2.3.9 Element Size

The EDMA controller does not use element-size and element-indexing. Instead, all transfers are defined in terms of all three dimensions: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, and EDMA_TPCC_CCNT_n[15:0] CCNT. An element-indexed transfer is logically achieved by programming EDMA_TPCC_ABCNT_n[15:0] ACNT to the size of the element and EDMA_TPCC_ABCNT_n[31:16] BCNT to the number of elements that need to be transferred. For example: If there are 16-bit audio data and 256 audio samples that must be transferred to a serial port, therefore the EDMA_TPCC_ABCNT_n[15:0] ACNT = 2 (2 bytes) and EDMA_TPCC_ABCNT_n[31:16] BCNT = 256.

12.2.4 Initiating a DMA Transfer

There are multiple ways to initiate a programmed data transfer using the EDMA_TPCC channel controller. Transfers on DMA channels are initiated by three sources.

They are listed as follows:

- **Event-triggered transfer request** (this is the typical usage of EDMA controller): A peripheral, system, or externally-generated event triggers a transfer request.
- **Manually-triggered transfer request:**The CPU manually triggers a transfer by writing a 1 to the corresponding bit in the event set registers (EDMA_TPCC_ESR / EDMA_TPCC_ESRH).
- **Chain-triggered transfer request:** A transfer is triggered on the completion of another transfer or sub-transfer.

Transfers on QDMA channels are initiated by two sources. They are as follows:

- **Auto-triggered transfer request:** Writing to the programmed trigger word triggers a transfer.
- **Link-triggered transfer requests:** Writing to the trigger word triggers the transfer when linking occurs.

12.2.4.1 DMA Channels

12.2.4.1.1 Event-Triggered Transfer Request

When an event is asserted from a peripheral or device pins, it gets latched in the corresponding bit of the event register (EDMA_TPCC_ER[31:0] E*n* = 1). For more information about peripheral events to EDMA events mapping, refer to *the device data manual*.
If the corresponding event in the event enable register (EDMA_TPCC_EER) is enabled (EDMA_TPCC_EER[31:0] E*n* = 1), then the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

If the PaRAM set is valid (not a NULL set), then a transfer request packet (TRP) is submitted to the EDMA_TPTC and the EDMA_TPCC_ER[31:0] E*n* bit is cleared. At this point, a new event can be safely received by the EDMA_TPCC.

If the PaRAM set associated with the channel is a NULL set (see Section 12.2.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_ER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register (EDMA_TPCC_EMR[31:0] E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include cleaning the event missed error before re-triggering the DMA channel.

When an event is received, the corresponding event bit in the event register is set (EDMA_TPCC_ER[31:0] E*n* = 1), regardless of the state of EDMA_TPCC_EER[31:0] E*n*. If the event is disabled when an external event is received (EDMA_TPCC_ER[31:0] E*n* = 1 and EDMA_TPCC_EER[31:0] E*n* = 0), the EDMA_TPCC_ER[31:0] E*n* bit remains set. If the event is subsequently enabled (EDMA_TPCC_EER[31:0] E*n* = 1), then the pending event is processed by the EDMA_TPCC and the TR is processed/submitted, after which the EDMA_TPCC_ER[31:0] E*n* bit is cleared.

If an event is being processed (prioritized or is in the event queue) and another sync event is received for the same channel prior to the original being cleared (EDMA_TPCC_ER[31:0] E*n* != 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EDMA_TPCC_EMR[31:0] E*n* = 1).

12.2.4.1.2 Manually-Triggered Transfer Request

The CPU or any peripheral device module initiates a DMA transfer by writing to the event set register EDMA_TPCC_ESR. Writing a 1 to an event bit in the EDMA_TPCC_ESR results in the event being prioritized/queued in the appropriate event queue, regardless of the state of the EDMA_TPCC_EER[31:0] E*n* bit. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 12.2.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_ER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA_TPCC_EMR[31:0] E*n* = 1 to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include clearing the event missed error before re-triggering the DMA channel.

If an event is being processed (prioritized or is in the event queue) and the same channel is manually set by a write to the corresponding channel bit of the event set register EDMA_TPCC_ESR[31:0] E*n* = 1 prior to the original being cleared EDMA_TPCC_ESR[31:0] E*n* = 0, then the second event is registered as a missed event in the corresponding bit of the event missed register EDMA_TPCC_EMR[31:0] E*n* = 1.

12.2.4.1.3 Chain-Triggered Transfer Request

Chaining is a mechanism by which the completion of one transfer automatically sets the event for another channel. When a chained completion code is detected, the value of which is dictated by the transfer completion code EDMA_TPCC_OPT_n[17:12] TCC of the PaRAM set associated with the channel, it results in the corresponding bit in the chained event register EDMA_TPCC_CER to be set EDMA_TPCC_CER[31:0] E[TCC] = 1).

Once a bit is set in EDMA_TPCC_CER, the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 12.2.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_CER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA_TPCC_EMR[31:0] E*n* = 1 to indicate that the event was discarded due to a null TR being serviced. In this case, the error condition must be cleared before the DMA channel can be re-triggered. Good programming practices might include clearing the event missed error before re-triggering the DMA channel.

If a chaining event is being processed (prioritized or queued) and another chained event is received for the same channel prior to the original being cleared EDMA_TPCC_CER[31:0] E*n* != 0), then the second chained event is registered as a missed event in the corresponding channel bit of the event missed register EDMA_TPCC_EMR[31:0] E*n* = 1.

NOTE: Chained event registers EDMA_TPCC_CER, event registers EDMA_TPCC_ER, and event set registers EDMA_TPCC_ESR operate independently. An event E*n* can be triggered by any of the trigger sources (event-triggered, manually-triggered, or chain-triggered).

12.2.4.2 QDMA Channels

12.2.4.2.1 Auto-Triggered and Link-Triggered Transfer Request

QDMA-based transfer requests are issued when a QDMA event gets latched in the QDMA event register EDMA_TPCC_QER[31:0] E*n* = 1. A bit corresponding to a QDMA channel is set in the QDMA event register EDMA_TPCC_QER when the following occurs:

- A CPU (or any device module) write occurs to a PaRAM address that is defined as a QDMA channel trigger word (programmed in the QDMA channel mapping register EDMA_TPCC_QCHMAPN_j for the particular QDMA channel and the QDMA channel is enabled via the QDMA event enable register EDMA_TPCC_QEER[31:0] E*n* = 1.
- EDMA_TPCC performs a link update on a PaRAM set address that is configured as a QDMA channel matches EDMA_TPCC_QCHMAPN_j settings and the corresponding channel is enabled via the QDMA event enable register EDMA_TPCC_QEER[31:0] E*n* = 1.

Once a bit is set in EDMA_TPCC_QER, the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If a bit is already set in EDMA_TPCC_QER[31:0] E*n* = 1 and a second QDMA event for the same QDMA channel occurs prior to the original being cleared, the second QDMA event gets captured in the QDMA event miss register EDMA_TPCC_QEMR[7:0] E*n* = 1.

12.2.4.3 Comparison Between DMA and QDMA Channels

The primary difference between DMA and QDMA channels is the event/channel synchronization.

QDMA events are either auto-triggered or link triggered. Auto-triggering allows QDMA channels to be triggered by CPU(s) with a minimum number of linear writes to PaRAM. Link triggering allows a linked list of transfers to be executed, using a single QDMA PaRAM set and multiple link PaRAM sets.

A QDMA transfer is triggered when a CPU (or other device modules) writes to the trigger word of the QDMA channel parameter set (auto-triggered) or when the EDMA_TPCC performs a link update on a PaRAM set that has been mapped to a QDMA channel (link triggered).

NOTE: The CPUs triggered (manually triggered) DMA channels, in addition to writing to the PaRAM set, it is required to write to the event set register EDMA_TPCC_ESR to kick-off the transfer.

QDMA channels are typically for cases where a single event accomplishes a complete transfer since the CPU (or other device modules) must reprogram some portion of the QDMA PaRAM set in order to re-trigger the channel. QDMA transfers are programmed with EDMA_TPCC_ABCNT_n[31:0] BCNT =1 and EDMA_TPCC_CCNT_n[15:0] CCNT = 1 for A-synchronized transfers, and EDMA_TPCC_CCNT_n[15:0] CCNT = 1 for AB-synchronized transfers.

Additionally, since linking is also supported (if EDMA_TPCC_OPT_n[3] STATIC = 0) for QDMA transfers, it allows to initiate a linked list of QDMAs, so when EDMA_TPCC copies over a link PaRAM set (including the write to the trigger word), the current PaRAM set mapped to the QDMA channel automatically recognizes as a valid QDMA event and initiate another set of transfers as specified by the linked set.

12.2.5 Completion of a DMA Transfer

A parameter set for a given channel is complete when the required number of transfer requests is submitted (based on receiving the number of synchronization events). The expected number of TRs for a non-null/non-dummy transfer is shown in Table 12-5 for both synchronization types along with state of the PaRAM set prior to the final TR being submitted. When the counts (EDMA_TPCC_ABCNT_n[31:0] BCNT and/or EDMA_TPCC_CCNT_n[15:0] CCNT) are this value, the next TR results in:

- Final chaining or interrupt codes sent by the transfer controllers (instead of intermediate).
- Link updates (linking to either null or another valid link set).

Table 12-5. Expected Number of Transfers for Non-Null Transfer

Sync Mode	Counts at time 0	Total # Transfers	Counts prior to final TR
A-synchronized	ACNT BCNT CCNT	(BCNT × CCNT) TRs of ACNT bytes each	EDMA_TPCC_ABCNT_n[31:0] BCNT == 1 && EDMA_TPCC_CCNT_n[15:0] CCNT == 1
AB-synchronized	ACNT BCNT CCNT	CCNT TRs for ACNT × BCNT bytes each	EDMA_TPCC_CCNT_n[15:0] CCNT == 1

The PaRAM OPT field must program with a specific transfer completion code TCC or EDMA_TPCC_OPT_n[17:12] TCC along with the other EDMA_TPCC_OPT_n fields ([22] TCCHEN, [20] TCINTEN, [23] ITCCHEN, and [21] ITCINTEN bits) to indicate whether the completion code is to be used for generating a chained event or/and for generating an interrupt upon completion of a transfer.

The specific EDMA_TPCC_OPT_n[17:12] TCC value (6-bit binary value) programmed dictates which of the 64-bits in the chain event register EDMA_TPCC_CER [TCC] and/or interrupt pending register EDMA_TPCC_IPR [TCC] is set.

It can selectively program whether the transfer controller sends back completion codes on completion of the final transfer request (TR) of a parameter set EDMA_TPCC_OPT_n[22] TCCHEN or EDMA_TPCC_OPT_n[20] TCINTEN, for all but the final transfer request (TR) of a parameter set EDMA_TPCC_OPT_n[23] ITCCHEN or EDMA_TPCC_OPT_n[21] ITCINTEN), or for all TRs of a parameter set (both). Refer to Section 12.2.8 *Chaining EDMA Channels* for details on chaining (intermediate/final chaining) and Section 12.2.9 *EDMA Interrupts* for details on intermediate/final interrupt completion.

A completion detection interface exists between the EDMA channel controller and transfer controller(s). This interface sends back information from the transfer controller to the channel controller to indicate that a specific transfer is completed. Completion of a transfer is used for generating chained events and/or generating interrupts to the CPU(s).

All DMA/QDMA PaRAM sets must also specify a link address value. For repetitive transfers such as ping-pong buffers, the link address value must point to another predefined PaRAM set. Alternatively, a non-repetitive transfer must set the link address value to the null link value. The null link value is defined as FFFFh. Refer to Section 12.2.3.7 *Linking Transfers* for more details.

NOTE: Any incoming events that are mapped to a null PaRAM set results in an error condition. The error condition must clear before the corresponding channel is used again. Refer to Section 12.2.3.5 *Dummy Versus Null Transfer Comparison*.

There are three ways the EDMA_TPCC gets updated/informed about a transfer completion: normal completion, early completion, and dummy/null completion. This applies to both chained events and completion interrupt generation.

12.2.5.1 Normal Completion

In normal completion mode EDMA_TPCC_OPT_n[11] TCCMODE = 0, the transfer or sub-transfer is considered to be complete when the EDMA channel controller receives the completion codes from the EDMA transfer controller. In this mode, the completion code to the channel controller is posted by the transfer controller after it receives a signal from the destination peripheral. Normal completion is typically used to generate an interrupt to inform the CPU that a set of data is ready for processing.

12.2.5.2 Early Completion

In early completion mode EDMA_TPCC_OPT_n[11] TCCMODE = 1, the transfer is considered to be complete when the EDMA channel controller submits the transfer request (TR) to the EDMA transfer controller. In this mode, the channel controller generates the completion code internally. Early completion is typically useful for chaining, as it allows subsequent transfers to be chained-triggered while the previous transfer is still in progress within the transfer controller, maximizing the overall throughput of the set of the transfers.

12.2.5.3 Dummy or Null Completion

This is a variation of early completion. Dummy or null completion is associated with a dummy set Section 12.2.3.4 or null set Section 12.2.3.3. In both cases, the EDMA channel controller does not submit the associated transfer request to the EDMA transfer controller(s). However, if the set (dummy/null) has the OPT field programmed to return completion code (intermediate/final interrupt/chaining completion), then it sets the appropriate bits in the interrupt pending registers EDMA_TPCC_IPR and EDMA_TPCC_IPRH or chained event register EDMA_TPCC_CER and EDMA_TPCC_CERH. The internal early completion path is used by the channel controller to return the completion codes internally (that is, EDMA_TPCC generates the completion code).

12.2.6 Event, Channel, and PaRAM Mapping

Several of the 64 DMA channels are tied to a specific hardware event, thus allowing events from device peripherals or external hardware (via the dma_evt[3:0] pins) to trigger transfers. A DMA channel typically requests a data transfer when it receives its event (apart from manually-triggered, chain-triggered, and other transfers). The amount of data transferred per synchronization event depends on the channel's configuration (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, EDMA_TPCC_CCNT_n[15:0] CCNT, etc.) and the synchronization type (A-synchronized or AB-synchronized).

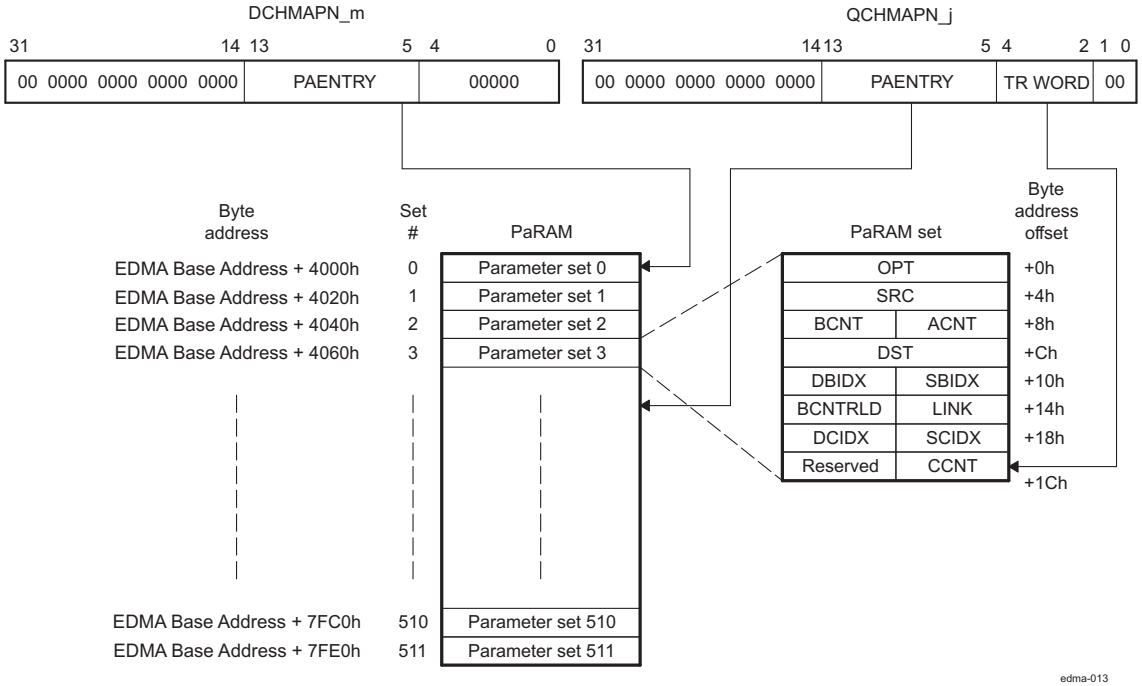
The association of an event to a channel is fixed within the EDMA Channel Controller, that is, each DMA channel has one specific event associated with it.

In an application, if a channel does not use the associated synchronization event or if it does not have an associated synchronization event (unused), that channel can be used for manually-triggered or chained-triggered transfers, for linking/reloading, or as a QDMA channel.

12.2.6.1 DMA Channel to PaRAM Mapping

The mapping between the DMA channel numbers and the PaRAM sets is programmable (see Table 12-1). The DMA channel mapping registers EDMA_TPCC_DCHMAPN_m in the EDMA_TPCC provide programmability that allows the DMA channels to be mapped to any of the PaRAM sets in the PaRAM memory map. Figure 12-11 illustrates the use of EDMA_TPCC_DCHMAPN_m. There is one EDMA_TPCC_DCHMAPN_m register per channel.

Figure 12-11. DMA Channel and QDMA Channel to PaRAM Mapping



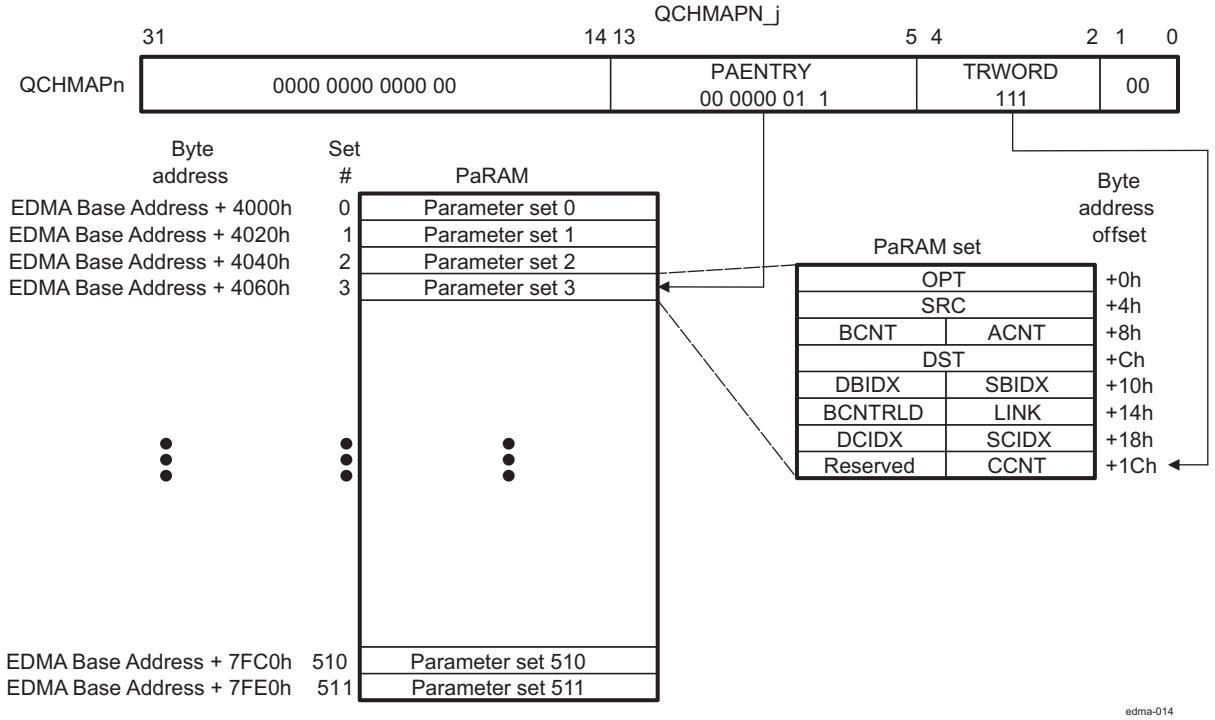
12.2.6.2 QDMA Channel to PaRAM Mapping

The mapping between the QDMA channels and the PaRAM sets is programmable. The QDMA channel mapping register EDMA_TPCC_QCHMAPN_j in the EDMA_TPCC allows to map the QDMA channels to any of the PaRAM sets in the PaRAM memory map. Figure 12-12 illustrates the use of EDMA_TPCC_QCHMAPN_j.

EDMA_TPCC_QCHMAPN_j[4:2] TRWORD bit-field allows to program the trigger word in the PaRAM set for the QDMA channel. A trigger word is one of the eight words in the PaRAM set. For a QDMA transfer to occur, a valid TR synchronization event for EDMA_TPCC is a write to the trigger word in the PaRAM set pointed to by EDMA_TPCC_QCHMAPN_j for a particular QDMA channel. By default, QDMA channels are mapped to PaRAM set 0.

It must appropriately re-map PaRAM set 0 before use.

Figure 12-12. QDMA Channel to PaRAM Mapping



12.2.7 EDMA Channel Controller Regions

The EDMA channel controller divides its address space into eight regions. Individual channel resources are assigned to a specific region, where each region is typically assigned to a specific device module uses the EDMA controller.

Application software can use regions or to ignore them altogether. It can be used active memory protection in conjunction with regions so that only a specific device module which uses the EDMA (for example, privilege identification) or privilege level (for example, user vs. supervisor) is allowed access to a given region, and thus to a given DMA or QDMA channel. This allows robust system-level DMA code where each EDMA initiator only modifies the state of the assigned resources. Memory protection is described in Section 12.2.10 *Memory Protection*.

12.2.7.1 Region Overview

The EDMA channel controller memory-mapped registers are divided in three main categories:

1. Global registers
2. Global region channel registers
3. Shadow region channel registers

The global registers are located at a single/fixed location in the EDMA_TPCC memory map. These registers control EDMA resource mapping and provide debug visibility and error tracking information.

The channel registers (including DMA, QDMA, and interrupt registers) are accessible via the global channel region address range, or in the shadow *n* channel region address range(s). For example, the event enable register EDMA_TPCC_EER is visible at the global address of EDMA Base Address + 1020h or region addresses of EDMA Base Address + 2020h for region 0, EDMA Base Address + 2220h for region 1, … EDMA Base Address + 2E20h for region 7.

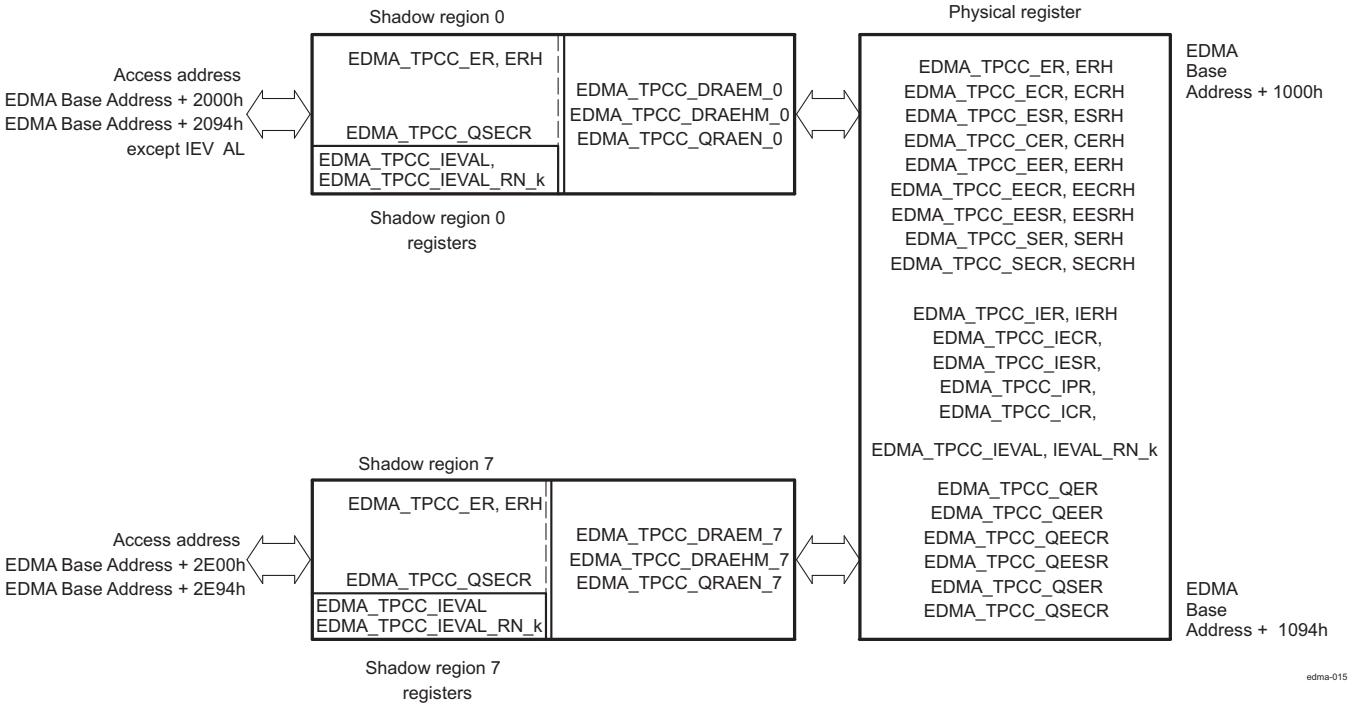
The DMA region access enable registers EDMA_TPCC_DRAEM_k and the QDMA region access enable registers EDMA_TPCC_QRAEN_k control the underlying control register bits that are accessible via the shadow region address space (except for EDMA_TPCC_IEVAL and EDMA_TPCC_IEVAL_RN_k registers). Table 12-6 lists the registers in the shadow region memory map. Refer to *EDMA_TPCC register mapping summary* for the complete global and shadow region memory maps.

Table 12-6. Shadow Region Registers

EDMA_TPCC_DRAEM_k	EDMA_TPCC_DRAEHM_k	EDMA_TPCC_QRAEN_k
EDMA_TPCC_ER	EDMA_TPCC_ERH	EDMA_TPCC_QER
EDMA_TPCC_ECR	EDMA_TPCC_ECRH	EDMA_TPCC_QEER
EDMA_TPCC_ESR	EDMA_TPCC_ESRH	EDMA_TPCC_QEECR
EDMA_TPCC_CER	EDMA_TPCC_CERH	EDMA_TPCC_QEESR
EDMA_TPCC_EER	EDMA_TPCC_EERH	
EDMA_TPCC_EECR	EDMA_TPCC_EECRH	
EDMA_TPCC_EESR	EDMA_TPCC_EESRH	
EDMA_TPCC_SER	EDMA_TPCC_SERH	
EDMA_TPCC_SECR	EDMA_TPCC_SECRH	
EDMA_TPCC_IER	EDMA_TPCC_IERH	
EDMA_TPCC_IECR	EDMA_TPCC_IECRH	
EDMA_TPCC_IESR	EDMA_TPCC_IESRH	
EDMA_TPCC_IPR	EDMA_TPCC_IPRH	
EDMA_TPCC_ICR	EDMA_TPCC_ICRH	
Register not affected by DRAE\DRAEH		
EDMA_TPCC_IEVAL		
EDMA_TPCC_IEVAL_RN_k		

Figure 12-13 illustrates the conceptual view of the regions.

Figure 12-13. Shadow Region Registers



12.2.7.2 Channel Controller Regions

There are eight EDMA shadow regions (and associated memory maps). Associated with each shadow region are a set of registers defining which channels and interrupt completion codes belong to that region. These registers are user-programmed per region to assign ownership of the DMA/QDMA channels to a region.

- EDMA_TPCC_DRAEM_k and EDMA_TPCC_DRAEHM_k: One register pair exists for each of the shadow regions. The number of bits in each register pair matches the number of DMA channels (64 DMA channels). These registers need to be programmed to assign ownership of DMA channels and interrupt (or EDMA_TPCC_OPT_n[17:12] TCC codes) to the respective region. Accesses to DMA and interrupt registers via the shadow region address view are filtered through the DRAEM/DRAEHM pair. A value of 1 in the corresponding EDMA_TPCC_DRAEM_k[31:0] / EDMA_TPCC_DRAEHM_k[31:0] bit implies that the corresponding DMA interrupt channel is accessible; a value of 0 in the corresponding EDMA_TPCC_DRAEM_k[31:0] / EDMA_TPCC_DRAEHM_k[31:0] bit forces writes to be discarded and returns a value of 0 for reads.
- EDMA_TPCC_QRAEN_k: One register exists for every region. The number of bits in each register matches the number of QDMA channels (4 QDMA channels). These registers must be programmed to assign ownership of QDMA channels to the respective region. To enable a channel in a shadow region using shadow region 0 EDMA_TPCC_QEER, the corresponding bits in QRAE must be set or writing into EDMA_TPCC_QEESR there will be no the desired effect.
- EDMA_TPCC_MPPAN_k and EDMA_TPCC_MPPAG: One register exists for every region. This register defines the privilege level, requestor, and types of accesses allowed to a region's memory-mapped registers.

It is typical for an application to have a unique assignment of QDMA/DMA channels (and, therefore, a given bit position) to a given region.

The use of shadow regions allows restricted access to EDMA resources (DMA channels, QDMA channels, TCC, interrupts) by tasks in a system by setting or clearing bits in the EDMA_TPCC_DRAEM_k / EDMA_TPCC_QRAEN_k registers.

If exclusive access to any given channel / TCC code is required for a region, then only that region's EDMA_TPCC_DRAEM_k / EDMA_TPCC_QRAEN_k have the associated bit set.

Example 12-1. Resource Pool Division Across Two Regions

This example illustrates a resource pool division across two regions, assuming region 0 must be allocated 16 DMA channels (0-15) and 1 QDMA channel (0) and 32 TCC codes (0-15 and 48-63).

Region 1 needs to be allocated 16 DMA channels (16-32) and the remaining 7 QDMA channels (1-7) and TCC codes (16-47).

EDMA_TPCC_DRAEM_k should be equal to the OR of the bits that are required for the DMA channels and the TCC codes:

```
Region 0: DRAEHM, DRAEM = 0xFFFF0000, 0x0000FFFF QRAEN = 0x0000001
Region 1: DRAEHM, DRAEM = 0x0000FFFF, 0xFFFF0000 QRAEN = 0x00000FE
```

12.2.7.3 Region Interrupts

In addition to the EDMA_TPCC global completion interrupt, there is an additional completion interrupt line that is associated with every shadow region. Along with the interrupt enable register EDMA_TPCC_IER, DRAEM acts as a secondary interrupt enable for the respective shadow region interrupts. Refer to *Hardware Request* for more information about EDMA Interrupts.

12.2.8 Chaining EDMA Channels

The channel chaining capability for the EDMA allows the completion of an EDMA channel transfer to trigger another EDMA channel transfer. The purpose is to allow the ability to chain several events through one event occurrence.

Chaining is different from linking (Section 12.2.3.7 *Linking Transfers*). The EDMA link feature reloads the current channel parameter set with the linked parameter set. The EDMA chaining feature does not modify or update any channel parameter set. It provides a synchronization event to the chained channel (see Section 12.2.4.1.3 *Chain-Triggered Transfer Request*).

Chaining is achieved at either final transfer completion or intermediate transfer completion, or both, of the current channel. Consider a channel *m* (DMA/QDMA) required to chain to channel *n*. Channel number *n* (0-63) needs to be programmed into the EDMA_TPCC_OPT_n[17:12] TCC bit-field of channel *m* channel options parameter (OPT) set.

- If final transfer completion chaining EDMA_TPCC_OPT_n[22] TCCHEN = 1 is enabled, the chain-triggered event occurs after the submission of the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).

- If intermediate transfer completion chaining EDMA_TPCC_OPT_n[23] ITCCHEN = 1 is enabled, the chain-triggered event occurs after every transfer request, except the last of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion chaining (EDMA_TPCC_OPT_n[22] TCCHEN = 1 and EDMA_TPCC_OPT_n[23] ITCCHEN = 1) are enabled, then the chain-trigger event occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 12-7 illustrates the number of chain event triggers occurring in different synchronized scenarios. Consider channel 31 programmed with EDMA_TPCC_ABCNT_n[15:0] ACNT = 3, EDMA_TPCC_ABCNT_n[31:16] BCNT = 4, EDMA_TPCC_CCNT_n[15:0] CCNT = 5, and EDMA_TPCC_OPT_n[17:12] TCC = 30.

Table 12-7. Chain Event Triggers

	(Number of chained event triggers on channel 30)	
Options	**A-Synchronized**	**AB-Synchronized**
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 0	1 (Owing to the last TR)	1 (Owing to the last TR)
EDMA_TPCC_OPT_n[22] TCCHEN = 0, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	19 (Owing to all but the last TR)	4 (Owing to all but the last TR)
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	20 (Owing to a total of 20 TRs)	5 (Owing to a total of 5 TRs)

12.2.9 EDMA Interrupts

The EDMA interrupts are divided into 2 categories: transfer completion interrupts and error interrupts.

There are nine region interrupts, eight shadow regions and one global region. The transfer completion interrupts are listed in Table 12-8. The transfer completion interrupts and the error interrupts from the transfer controllers are all routed to the device interrupt controllers INTCs through the inputs of the IRQ_CROSSBAR module.

Table 12-8. EDMA Transfer Completion Interrupts

Name	Description
EDMA_TPCC_INT0	EDMA_TPCC Transfer Completion Interrupt Shadow Region 0
EDMA_TPCC_INT1	EDMA_TPCC Transfer Completion Interrupt Shadow Region 1
EDMA_TPCC_INT2	EDMA_TPCC Transfer Completion Interrupt Shadow Region 2
EDMA_TPCC_INT3	EDMA_TPCC Transfer Completion Interrupt Shadow Region 3
EDMA_TPCC_INT4	EDMA_TPCC Transfer Completion Interrupt Shadow Region 4
EDMA_TPCC_INT5	EDMA_TPCC Transfer Completion Interrupt Shadow Region 5
EDMA_TPCC_INT6	EDMA_TPCC Transfer Completion Interrupt Shadow Region 6
EDMA_TPCC_INT7	EDMA_TPCC Transfer Completion Interrupt Shadow Region 7

Table 12-9. EDMA Error Interrupts

Name	Description
EDMA_TPCC_ERRINT	EDMA_TPCC Error Interrupt
EDMA_TPCC_MPINT	EDMA_TPCC Memory Protection Interrupt
EDMA_TC0_ERRINT	TC0 Error Interrupt
EDMA_TC1_ERRINT	TC1 Error Interrupt

12.2.9.1 Transfer Completion Interrupts

The EDMA_TPCC is responsible for generating transfer completion interrupts to the CPU(s) (and other EDMA masters). The EDMA generates a single completion interrupt per shadow region, as well as one for the global region on behalf of all 64 channels. The various control registers and bit fields facilitate EDMA interrupt generation.

The software architecture must either use the global interrupt or the shadow interrupts, but not both.

The transfer completion code EDMA_TPCC_OPT_n[17:12] TCC value is directly mapped to the bits of the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH.

For example, if EDMA_TPCC_OPT_n[17:12] TCC = 10 0001b, EDMA_TPCC_IPRH[1] is set after transfer completion, and results in interrupt generation to the CPU(s) if the completion interrupt is enabled for the CPU. See Section 12.2.9.1.1 *Enabling Transfer Completion Interrupts* for details about enabling EDMA transfer completion interrupts.

When a completion code is returned (as a result of early or normal completions), the corresponding bit in EDMA_TPCC_IPR / EDMA_TPCC_IPRH registers is set if transfer completion interrupt (final/intermediate) is enabled in the channel options parameter (OPT) for a PaRAM set associated with the transfer.

Table 12-10. Transfer Complete Code (TCC) to EDMA_TPCC Interrupt Mapping

TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPR Bit Set	TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPRH Bit Set[1]
0	EDMA_TPCC_IPR[0]	20h	EDMA_TPCC_IPR[32] / EDMA_TPCC_IPRH[0]
1	EDMA_TPCC_IPR[1]	21h	EDMA_TPCC_IPR[33] / EDMA_TPCC_IPRH[1]
2h	EDMA_TPCC_IPR[2]	22h	EDMA_TPCC_IPR[34] / EDMA_TPCC_IPRH[2]
3h	EDMA_TPCC_IPR[3]	23h	EDMA_TPCC_IPR[35] / EDMA_TPCC_IPRH[3]
4h	EDMA_TPCC_IPR[4]	24h	EDMA_TPCC_IPR[36] / EDMA_TPCC_IPRH[4]
...	...	...	...
1Eh	EDMA_TPCC_IPR[30]	3Eh	EDMA_TPCC_IPR[62] / EDMA_TPCC_IPRH[30]
1Fh	EDMA_TPCC_IPR[31]	3Fh	EDMA_TPCC_IPR[63] / EDMA_TPCC_IPRH[31]

(1) Bit fields EDMA_TPCC_IPR [32-63] correspond to bits 0 to 31 in EDMA_TPCC_IPRH, respectively.

The transfer completion code (TCC) can program to any value for a DMA/QDMA channel. A direct relation between the channel number and the transfer completion code value does not need to exist. This allows multiple channels having the same transfer completion code value to cause a CPU to execute the same interrupt service routine (ISR) for different channels.

If the channel is used in the context of a shadow region and it intends for the shadow region interrupt to be asserted, then ensure that the bit corresponding to the TCC code is enabled in EDMA_TPCC_IER / EDMA_TPCC_IERH and in the corresponding shadow region's DMA region access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k).

Interrupt generation can be enabled at either final transfer completion or intermediate transfer completion, or both. Consider channel *m* as an example.

- If the final transfer interrupt (EDMA_TPCC_OPT_n[20] TCINTEN = 1) is enabled, the interrupt occurs after the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).

- If the intermediate transfer interrupt (EDMA_TPCC_OPT_n[21] ITCINTEN = 1) is enabled, the interrupt occurs after every transfer request, except the last TR of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion interrupts (EDMA_TPCC_OPT_n[20] TCINTEN = 1, and EDMA_TPCC_OPT_n[21] ITCINTEN = 1) are enabled, then the interrupt occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 12-11 shows the number of interrupts that occur in different synchronized scenarios. Consider channel 31, programmed with ABCNT_n[15:0] ACNT = 3, EDMA_TPCC_ABCNT_n[31:16] BCNT = 4, EDMA_TPCC_CCNT_n[15:0]CCNT = 5, and EDMA_TPCC_OPT_n[17:12] TCC = 30.

Table 12-11. Number of Interrupts

Options	A-Synchronized	AB-Synchronized
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 0	1 (Last TR)	1 (Last TR)
EDMA_TPCC_OPT_n[20] TCINTEN = 0, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	19 (All but the last TR)	4 (All but the last TR)
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	20 (All TRs)	5 (All TRs)

12.2.9.1.1 Enabling Transfer Completion Interrupts

For the EDMA channel controller to assert a transfer completion to the external environment, the interrupts must be enabled in the EDMA_TPCC. This is in addition to setting up the EDMA_TPCC_OPT_n[20] TCINTEN and EDMA_TPCC_OPT_n[21] ITCINTEN bits of the associated PaRAM set.

The EDMA channel controller has interrupt enable registers EDMA_TPCC_IER / EDMA_TPCC_IERH and each bit location in EDMA_TPCC_IER / EDMA_TPCC_IERH serves as a primary enable for the corresponding interrupt pending registers EDMA_TPCC_IPR / EDMA_TPCC_IPRH.

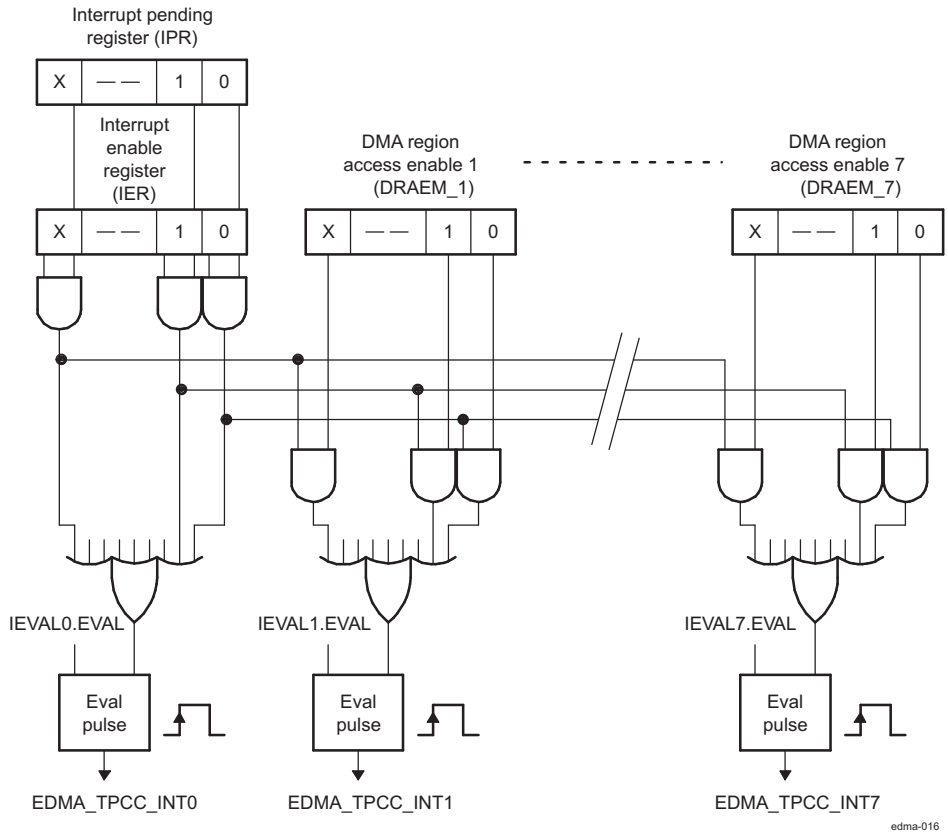
All of the interrupt registers (EDMA_TPCC_IER, EDMA_TPCC_IESR, EDMA_TPCC_IECR, and EDMA_TPCC_IPR) are either manipulated from the global DMA channel region, or by the DMA channel shadow regions. The shadow regions provide a view to the same set of physical registers that are in the global region.

The EDMA channel controller has a hierarchical completion interrupt scheme that uses a single set of interrupt pending registers EDMA_TPCC_IPR / EDMA_TPCC_IPRH and single set of interrupt enable registers EDMA_TPCC_IER / EDMA_TPCC_IERH. The programmable DMA region access enable registers EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k provides a second level of interrupt masking. The global region interrupt output is gated based on the enable mask that is provided by EDMA_TPCC_IER / EDMA_TPCC_IERH, see Figure 12-14

The region interrupt outputs are gated by EDMA_TPCC_IER and the specific EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k associated with the region.

Figure 12-14 shows the Interrupt diagram of the EDMA controller.

Figure 12-14. Interrupt Diagram



The EDMA_TPCC generates the transfer completion interrupts that are associated with each shadow region, the following conditions must be true:

- EDMA_TPCC_INT0: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_0[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_0[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_0[31] E63)
- EDMA_TPCC_INT1: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_1[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_1[1] E1) | …| (EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_1[31] E63)
- EDMA_TPCC_INT2: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_2[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_2[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_2[31] E63)....
- Up to EDMA_TPCC_INT7: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_7[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_7[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_7[31] E63)

NOTE: The EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k for all regions are expected to be set up at system initialization and to remain static for an extended period of time. The interrupt enable registers are used for dynamic enable/disable of individual interrupts.

Because there is no relation between the EDMA_TPCC_OPT_n[17:12] TCC value and the DMA/QDMA channel, it is possible, the DMA channel 0 to have the EDMA_TPCC_OPT_n[17:12] TCC = 63 in its associated PaRAM set. This mean that if a transfer completion interrupt is enabled (EDMA_TPCC_OPT_n[20] TCINTEN or EDMA_TPCC_OPT_n[21] ITCINTEN is set), then based on the TCC value, EDMA_TPCC_IPRH[31] E63 is set up on completion. For proper channel operations and interrupt generation using the shadow region map - program the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k that is associated with the shadow region to have read/write access to both bit 0 (corresponding to channel 0) and bit 63 (corresponding to EDMA_TPCC_IPRH bit that is set upon completion).

12.2.9.1.2 Clearing Transfer Completion Interrupts

Transfer completion interrupts that are latched to the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH) are cleared by writing a 1 to the corresponding bit in the interrupt pending clear register (EDMA_TPCC_ICR / EDMA_TPCC_ICRH). For example, a write of 1 to EDMA_TPCC_ICR[0] E0 clears a pending interrupt in EDMA_TPCC_IPR[0] E0.

If an incoming transfer completion code TCC (EDMA_TPCC_OPT_n[17:12] TCC) gets latched to a bit in EDMA_TPCC_IPR / EDMA_TPCC_IPRH, then additional bits that get set due to a subsequent transfer completion does not result in asserting the EDMA_TPCC completion interrupt. In order for the completion interrupt to be pulsed, the required transition is from a state where no enabled interrupts are set to a state where at least one enabled interrupt is set.

12.2.9.2 EDMA Interrupt Servicing

Upon completion of a transfer (early or normal completion), the EDMA channel controller sets the appropriate bit in the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH), as the transfer completion codes specify. If the completion interrupts are appropriately enabled, then the CPU enters the interrupt service routine (ISR) when the completion interrupt is asserted.

After servicing the interrupt, the ISR should clear the corresponding bit in EDMA_TPCC_IPR/EDMA_TPCC_IPRH, thereby enabling recognition of future interrupts. The EDMA_TPCC only asserts additional completion interrupts when all EDMA_TPCC_IPR / EDMA_TPCC_IPRH bits clear.

When one interrupt is serviced many other transfer completions may result in additional bits being set in EDMA_TPCC_IPR / EDMA_TPCC_IPRH, thereby resulting in additional interrupts. Each of the bits in EDMA_TPCC_IPR / EDMA_TPCC_IPRH may need different types of service therefore, the ISR must check all pending interrupts and continue until all of the posted interrupts are serviced appropriately.

Examples of pseudo code for a CPU interrupt service routine for an EDMA_TPCC completion interrupt are shown in Example 12-2 and Example 12-3.

The ISR routine in Example 12-2 is more exhaustive and incurs a higher latency.

Example 12-2. Interrupt Servicing

The pseudo code:

1. Reads the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH.
2. Performs the operations needed.
3. Writes to the interrupt pending clear register EDMA_TPCC_ICR / EDMA_TPCC_ICRH to clear the corresponding EDMA_TPCC_IPR / EDMA_TPCC_IPRH bit(s).
4. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH again:
   a. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is not equal to 0, repeat from step 2 (implies occurrence of new event between step 2 to step 4).

Example 12-2. Interrupt Servicing (continued)

b. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is equal to 0, assure that all of the enabled interrupts are inactive.

NOTE: An event may occur during step 4 while the EDMA_TPCC_IPR / EDMA_TPCC_IPRH bits are read as 0 and the application is still in the interrupt service routine. If this happens, a new interrupt is recorded in the device interrupt controller and a new interrupt generates as soon as the application exits in the interrupt service routine.

Example 12-3 is less rigorous, with less burden on the software in polling for set interrupt bits, but can occasionally cause a race condition as mentioned above.

Example 12-3. Interrupt Servicing

If any enabled and pending (possibly lower priority) interrupts are left, force the interrupt logic to reassert the interrupt pulse by setting the EDMA_TPCC_IEVAL[0] EVAL bit in the interrupt evaluation register.

The pseudo code is as follows:

1. Enters ISR.
2. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH.
3. For the condition that is set in EDMA_TPCC_IPR / EDMA_TPCC_IPRH:
   a. Service interrupt as the application requires.
   b. Clear the bit for serviced conditions (others may still be set, and other transfers may have resulted in returning the TCC to EDMA_TPCC after step 2).
4. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH prior to exiting the ISR:
   a. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is equal to 0, then exit the ISR.
   b. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is not equal to 0, then set EDMA_TPCC_IEVAL so that upon exit of ISR, a new interrupt triggers if any enabled interrupts are still pending.

12.2.9.3 Interrupt Evaluation Operations

The EDMA_TPCC has interrupt evaluate registers EDMA_TPCC_IEVAL that exist in the global region and in each shadow region. The registers in the shadow region are the only registers in the DMA channel shadow region memory map that are not affected by the settings for the DMA region access enable registers EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k. Writing a 1 to the EDMA_TPCC_IEVAL[0] EVAL bit in the registers that are associated with a particular shadow region results in pulsing the associated region interrupt (global or shadow), if any enabled interrupt (via EDMA_TPCC_IER / EDMA_TPCC_IERH) is still pending EDMA_TPCC_IPR / EDMA_TPCC_IPRH. This register assures that the CPU does not miss the interrupts (or the EDMA master associated with the shadow region) if the software architecture chooses not to use all interrupts. Refer to Example 12-3 about the use of EDMA_TPCC_IEVAL in the EDMA interrupt service routine (ISR).

Similarly an error evaluation register EDMA_TPCC_EEVAL exists in the global region. Writing a 1 to the EDMA_TPCC_EEVAL[0] EVAL bit causes the pulsing of the error interrupt if any pending errors are in EDMA_TPCC_EMR / EDMA_TPCC_EMRH, EDMA_TPCC_QEMR, or EDMA_TPCC_CCERR. See Section 12.2.9.4 *Error Interrupts* for additional information regarding error interrupts.

NOTE: While using EDMA_TPCC_IEVAL for shadow region completion interrupts, check that the EDMA_TPCC_IEVAL operated upon is from that particular shadow region memory map.

12.2.9.4 Error Interrupts

The EDMA_TPCC error registers provide the capability to differentiate error conditions (event missed, threshold exceed, etc.). Additionally, setting the error bits in these registers results in asserting the EDMA_TPCC error interrupt. If the EDMA_TPCC error interrupt is enabled in the device interrupt controller(s), then it allows the CPU(s) to handle the error conditions.

The EDMA_TPCC has a single error interrupt (EDMA_TPCC_ERRINT) that is asserted for all EDMA_TPCC error conditions. There are four conditions that cause the error interrupt:

- DMA missed events: for all 64 DMA channels. DMA missed events are latched in the event missed registers EDMA_TPCC_EMR / EDMA_TPCC_EMRH.
- QDMA missed events: for all 8 QDMA channels. QDMA missed events are latched in the QDMA event missed register EDMA_TPCC_QEMR.
- Threshold exceed: for all event queues. These are latched in EDMA_TPCC error register EDMA_TPCC_CCERR.
- TCC error: for outstanding transfer requests that are expected to return completion code EDMA_TPCC_OPT_n[22] TCCHEN or EDMA_TPCC_OPT_n[23] TCINTEN bit is set to 1, exceeding the maximum limit of 63. This is also latched in the EDMA_TPCC error register EDMA_TPCC_CCERR.

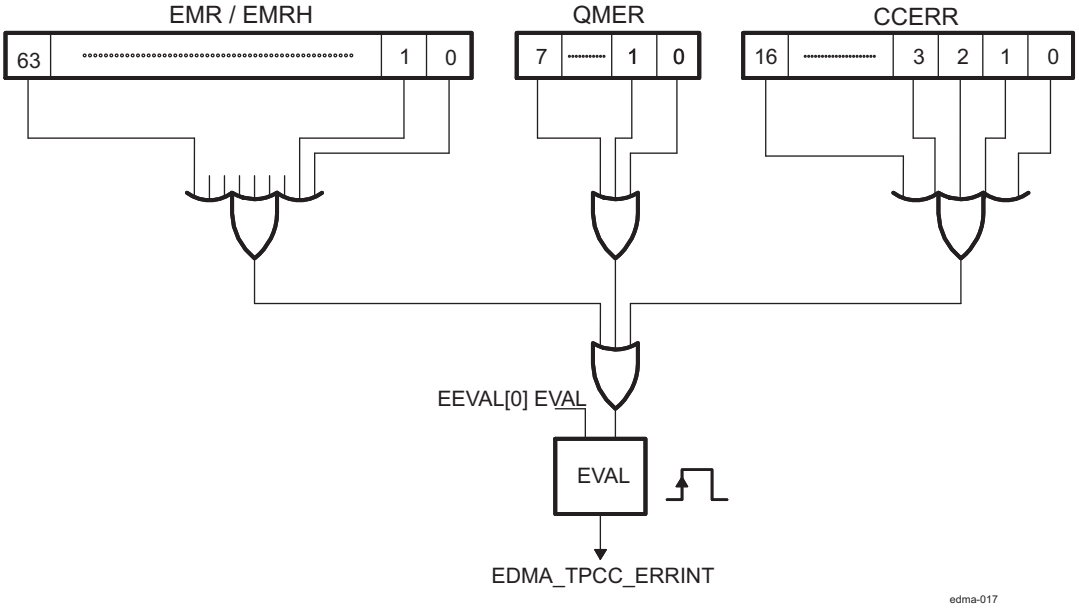
Figure 12-15 illustrates the EDMA_TPCC error interrupt generation operation.

If any of the bits are set in the error registers due to any error condition, the EDMA_TPCC_ERRINT is always asserted, as there are no enables for masking these error events. Similar to transfer completion interrupts (EDMA_TPCC_INT), the error interrupt also only pulses when the error interrupt condition transitions from no errors being set to at least one error being set. If additional error events are latched prior to the original error bits clearing, the EDMA_TPCC does not generate additional interrupt.

To reduce the burden on the software, there is an error evaluate register EDMA_TPCC_EEVAL that allows re-evaluation of pending set error events/bits, similar to the interrupt evaluate register EDMA_TPCC_IEVAL. Unlike the EDMA_TPCC_IEVAL functionality, the EDMA_TPCC_EEVAL register must be written with ‘1’ after any error interrupts are serviced (even when all pending errors are cleared) in order for subsequent errors to trigger a new interrupt.

NOTE: It is good practice to enable the error interrupt in the device interrupt controller and to associate an interrupt service routine with it to address the various error conditions appropriately. Doing so puts less burden on the software (polling for error status), it provides a good debug mechanism for unexpected error conditions.

Figure 12-15. Error Interrupt Operation



12.2.10 Memory Protection

The EDMA channel controller supports two kinds of memory protection: active and proxy.

12.2.10.1 Active Memory Protection

Active memory protection is a feature that allows or prevents read and write accesses to the EDMA_TPCC registers. Active memory protection is achieved by a set of memory protection permissions attribute EDMA_TPCC_MPPAN_k registers.

The EDMA_TPCC register map is divided into three categories:

- a global region.
- a global channel region.
- eight shadow regions.

Each shadow region consists of the respective shadow region registers and the associated PaRAM. For more detailed information regarding the contents of a shadow region, refer to *EDMA_TPCC Registers Mapping Summary*.

Each of the eight shadow regions has an associated EDMA_TPCC_MPPAN_k registers that defines the specific requestor(s) and types of requests that are allowed to the regions resources.

The global channel region is also protected with a memory-mapped register EDMA_TPCC_MPPAG. The EDMA_TPCC_MPPAG applies to the global region and to the global channel region, except the other EDMA_TPCC_MPPAN_k registers themselves.

Table 12-12 shows the accesses that are allowed or not allowed to the EDMA_TPCC_MPPAG and EDMA_TPCC_MPPAN_k. The active memory protection uses the EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID attributes of the EDMA peripheral modules. TheEDMA_TPCC_OPT_n[31] PRIV is the privilege level (i.e., user vs. supervisor).

The EDMA_TPCC_OPT_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an EDMA peripheral modules.

Table 12-12. Allowed Accesses

Access	Supervisor	User
Read	Yes	Yes
Write	Yes	No

Table 12-13 describes the EDMA_TPCC_MPPAN_k register mapping for the shadow regions (which includes shadow region registers and PaRAM addresses).

The region-based EDMA_TPCC_MPPAN_k registers are used to protect accesses to the DMA shadow regions and the associated region PaRAM. Because there are eight regions, there are eight EDMA_TPCC_MPPAN_k region registers (MPPA[0-7]).

Table 12-13. MPPA Registers to Region Assignment

Register	Registers Protect	Address Range	PaRAM Protect[1]	Address Range
EDMA_TPCC_MPPAG	Global Range	0000h-1FFCh	N/A	N/A
EDMA_TPCC_MPPAN_k. MPPAN_0	DMA Shadow 0	2000h-21FCh	1st octant	4000h-47FCh
MPPAN_1	DMA Shadow 1	2200h-23FCh	2nd octant	4800h-4FFCh
MPPAN_2	DMA Shadow 2	2400h-25FCh	3rd octant	5000h-57FCh
MPPAN_3	DMA Shadow 3	2600h-27FCh	4th octant	5800h-5FFCh
MPPAN_4	DMA Shadow 4	2800h-29FCh	5th octant	6000h-67FCh
MPPAN_5	DMA Shadow 5	2A00h-2BFCh	6th octant	6800h-6FFCh
MPPAN_6	DMA Shadow 6	2C00h-2DFCh	7th octant	7000h-77FCh
MPPAN_7	DMA Shadow 7	2E00h-2FFCh	8th octant	7800h-7FFCh

(1) The PARAM region is divided into 8 regions referred to as an octant.

Example *Access denied.*

Write access to shadow region 7's event enable set register EDMA_TPCC_EESR:

1. The original value of the event enable register EDMA_TPCC_EER at address offset 0x1020 is 0x0.
2. The EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[7] NS is set to prevent user level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 0, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 0), but it allows supervisor level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1) with a privilege ID of 0. (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0 attempts to perform a user-level write of a value of 0xFF00FF00 to shadow region 7's event enable set register EDMA_TPCC_EESR at address offset 0x2E30.

NOTE: The EDMA_TPCC_EER is a read-only register and the only way that write to it is by writing to the EDMA_TPCC_EESR. There is only one physical register for EDMA_TPCC_EER, EDMA_TPCC_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 0, though the privilege ID of the write access is set to 0, the access is not allowed and the EDMA_TPCC_EER is not written too.

Table 12-14. Example Access Denied

Register	Value	Description
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Value in EDMA_TPCC_EER to begin with.
EDMA_TPCC_EESR (offset 0x2E30)	0xFF00 FF00 ↓	Value attempted to be written to shadow region 7's EDMA_TPCC_EESR. This is done by an EDMA connected device module with a privilege level of User and Privilege ID of 0.
EDMA_TPCC_MPPAN_k (offset 0x082C)	0x0000 04B0	Memory Protection Filter EDMA_TPCC_MPPAN_k[10] AID0 = 1, EDMA_TPCC_MPPAN_k[1] UW = 0, EDMA_TPCC_MPPAN_k[2] UR = 0, EDMA_TPCC_MPPAN_k[4] SW = 1, EDMA_TPCC_MPPAN_k[5] SR = 1.
	X	Access Denied
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Final value of EDMA_TPCC_EER

Example *Access Allowed*

Write access to shadow region 7's event enable set register EDMA_TPCC_EESR:

1. The original value of the event enable register EDMA_TPCC_EER at address offset 0x1020 is 0x0.
2. The EDMA_TPCC_MPPAN_k.EDMA_TPCC_MPPAN_7 is set to allow user-level accesses (EDMA_TPCC_MPPAN_k.EDMA_TPCC_MPPAN_7[1] UW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 1) and supervisor-level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1) with a privilege ID of 0. (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0, attempts to perform a user-level write of a value of 0xABCD0123 to shadow region 7's event enable set register EDMA_TPCC_EESR at address offset 0x2E30.

NOTE: The EDMA_TPCC_EER is a read-only register and the only way that write to it is by writing to the EDMA_TPCC_EESR. There is only one physical register for EDMA_TPCC_EER, EDMA_TPCC_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 1 and EDMA_TPCC_MPPAN_k. MPPAN_7[10] AID0 = 1, the user-level write access is allowed.
5. The accesse to shadow region registers are masked by their respective EDMA_TPCC_DRAEM_k

register. In this example, the EDMA_TPCC_DRAEM_k. EDMA_TPCC_DRAEM_7 is set of 0x9FF00FC2.

6. The value finally written to EDMA_TPCC_EER is 0x8BC00102.

Table 12-15. Example Access Allowed

Register	Value	Description
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Value in EER to begin with.
EDMA_TPCC_EESR (offset 0x2E30)	0xFF00 FF00	Value attempted to be written to shadow region 7's EESR. This is done by an EDMA peripheral module with a privilege level of User and Privilege ID of 0.
EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7 (offset 0x082C)	0x0000 04B3	Memory Protection Filter EDMA_TPCC_MPPAN_k[10] AID = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1.
	√ ↓	Access allowed.
EDMA_TPCC_DRAEM_k. EDMA_TPCC_DRAEM_7 (offset 0x0378)	0x9FF0 0FC2 ↓	DMA Region Access Enable Filter
EDMA_TPCC_EESR (offset 0x2E30)	0x8BC0 0102 ↓	Value written to shadow region 7's EESR. This is done by an EDMA peripheral module with a privilege level of User and a Privilege ID of 0.
EDMA_TPCC_EER (offset 0x1020)	↓ 0xBC0 0102	Final value of EER.

12.2.10.2 Proxy Memory Protection

Proxy memory protection allows an EDMA transfer programmed by a given peripheral module connected to EDMA, to have its permissions travel with the transfer through the EDMA_TPTC. The permissions travel along with the read transactions to the source and the write transactions to the destination endpoints. The EDMA_TPCC_OPT_n[31] PRIV bit and EDMA_TPCC_OPT_n[27:24] PRIVID bit is set with the peripheral module's PRIV value and PRIVID values, respectively, when any part of the PaRAM set is written.

The EDMA_TPCC_OPT_n[31] PRIV is the privilege level (i.e., user vs. supervisor). The EDMA_TPCC_OPT_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an peripheral module connected to EDMA.

These options are part of the TR that are submitted to the transfer controller. The transfer controller uses the above values on their respective read and write command bus so that the target endpoints can perform memory protection checks based on these values.

Consider a parameter set that is programmed by a CPU in user privilege level for a simple transfer with the source buffer on an L2 page and the destination buffer on an L1D page. The EDMA_TPCC_OPT_n[31] PRIV is 0 for user-level and the CPU has a EDMA_TPCC_OPT_n[27:24] PRIVID to 0.

The PaRAM set is shown in Figure 12-16.

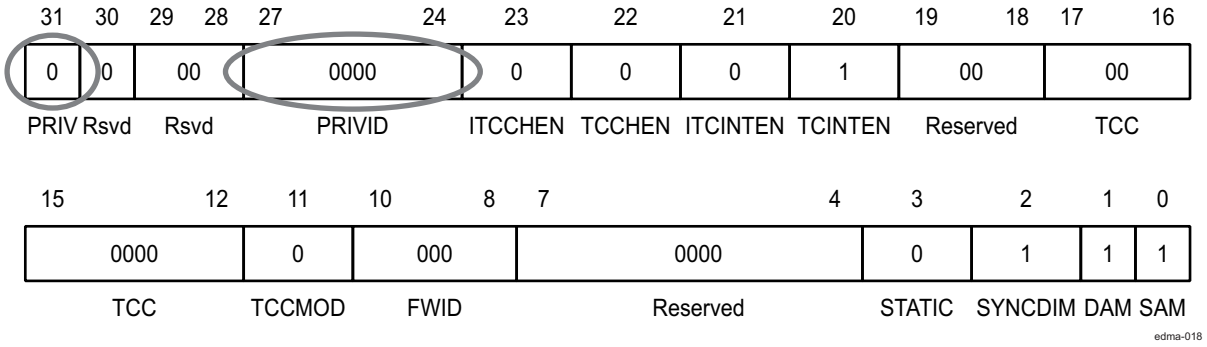
Figure 12-16. PaRAM Set Content for Proxy Memory Protection Example

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0007h		Channel Options Parameter (OPT)	
009F 0000h		Channel Source Address (SRC)	
0001h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
00F0 7800h		Channel Destination Address (DST)	
0001h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

Figure 12-17. Channel Options Parameter (OPT) Example

(b) Channel Options Parameter (OPT_n) Content



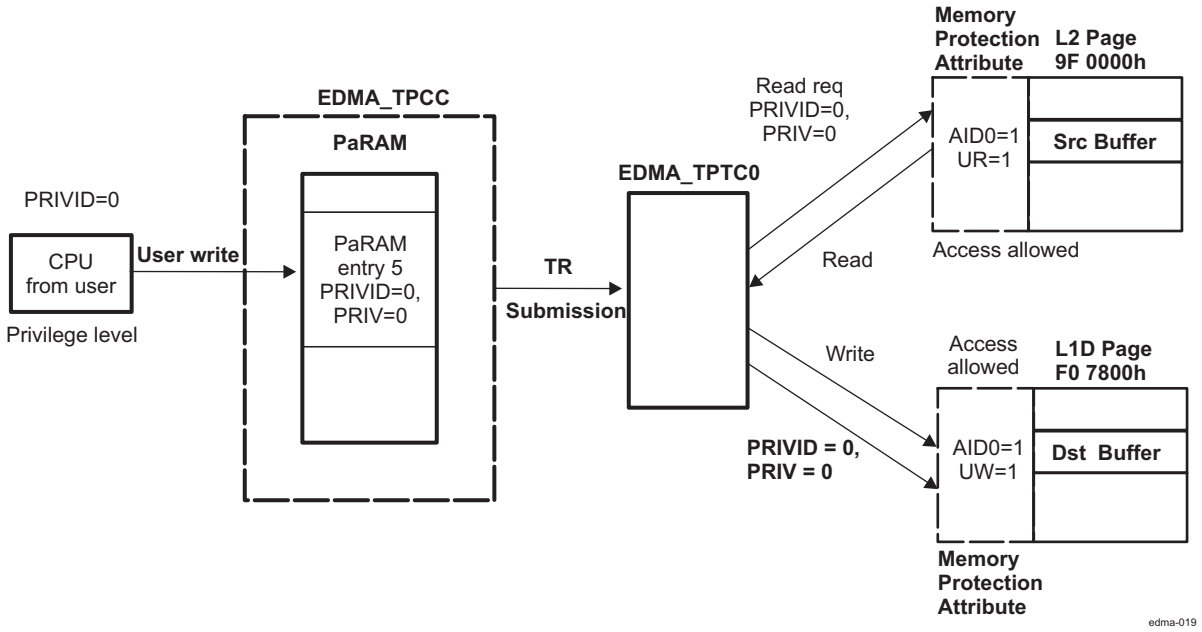
The EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID information travels along with the read and write requests that are issued to the source and destination memories.

For example, if the access attributes that are associated with the L2 page with the source buffer only allow supervisor read, write accesses EDMA_TPCC_MPPAN_k[4] SW and EDMA_TPCC_MPPAN_k[5] SR, the user-level read request above is refused. Similarly, if the access attributes that are associated with the L1D page with the destination buffer only allow supervisor read and write accesses (EDMA_TPCC_MPPAN_k[4] SW, EDMA_TPCC_MPPAN_k[5] SR), the user-level write request above is refused. For the transfer to succeed, the source and destination pages must have user-read and user-write permissions, respectively, along with allowing accesses from a PRIVID = 0.

Because the privilege level and privilege identification travel with the read and write requests, EDMA acts as a proxy.

Figure 12-18 illustrates the propagation of EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID at the boundaries of all the interacting entities (CPU, EDMA_TPCC, EDMA_TPTCs, and slave memories).

Figure 12-18. Proxy Memory Protection Example



12.2.11 Event Queue(s)

Event queues are a part of the EDMA channel controller. Event queues form the interface between the event detection logic in the EDMA_TPCC and the transfer request (TR) submission logic of the EDMA_TPCC. Each queue is 16 entries deep. Each event queue can queue a maximum of 16 events. If there are more than 16 events, then the events that cannot find a place in the event queue remain set in the associated event register and the CPU does not stall.

There are two event queues for the device: Queue0, Queue1. Events in Queue0 result in submission of its associated transfer requests (TRs) to TC0. The transfer requests that are associated with events in Queue1 are submitted to TC1.

An event that wins prioritization against other DMA and/or QDMA pending events is placed at the tail of the appropriate event queue. Each event queue is serviced in FIFO order. Once the event reaches the head of its queue and the corresponding transfer controller is ready to receive another TR, the event is de-queued and the PaRAM set corresponding to the de-queued event is processed and submitted as a transfer request packet (TRP) to the associated EDMA transfer controller.

Queue0 has highest priority and Queue1 has the lowest priority, if Queue0 and Queue1 both have at least one event entry and if both TC0 and TC1 can accept transfer requests, then the event in Queue0 is de-queued first and its associated PaRAM set is processed and submitted as a transfer request (TR) to TC0.

Refer to Section 12.2.11.4 for system-level performance considerations. All of the event entries in all of the event queues are software readable (not writeable) by accessing the event entry registers EDMA_TPCC_Q0E_p and EDMA_TPCC_Q1E_p. Each event entry register characterizes the queued event in terms of the type of event (manual, event, chained or auto-triggered) and the event number. Refer to *EDMA_TPCC Register Description* for EDMA_TPCC_Q0E_p / EDMA_TPCC_Q1E_p descriptions of the bit fields.

12.2.11.1 DMA/QDMA Channel to Event Queue Mapping

Each of the 64 DMA channels and eight QDMA channels are programmed independently to map to a specific queue, using the DMA queue number register EDMA_TPCC_DMAQNUMN_k and the QDMA queue number register EDMA_TPCC_QDMAQNUM. The mapping of DMA/QDMA channels is critical to achieving the desired performance level for the EDMA and most importantly, in meeting real-time deadlines. Refer to Section 12.2.11.4 *System-level Performance Considerations*.

NOTE: If an event is ready to be queued and both the event queue and the EDMA transfer controller that is associated to the event queue are empty, then the event bypasses the event queue, and moves the PaRAM processing logic, and eventually to the transfer request submission logic for submission to the EDMA_TPTC. In this case, the event is not logged in the event queue status registers.

12.2.11.2 Queue RAM Debug Visibility

There are two event queues and each queue has 16 entries. These 16 entries are managed in a circular FIFO. There is a queue status register EDMA_TPCC_QSTATN_i associated with each queue. These along with all of the 16 entries per queue can be read via registers EDMA_TPCC_QSTATN_i and Q0E_p / Q1E_p, respectively.

These registers provide user visibility.

The event queue entry register (Q*x*E*y* Q0E_p / Q1E_p) uniquely identifies the specific event type (event-triggered, manually-triggered, chain-triggered, and QDMA events) along with the event number (for all DMA/QDMA event channels) that are in the queue or have been de-queued (passed through the queue).

Each of the 16 entries in the event queue are read using the EDMA_TPCC memory-mapped register. To see the history of the last 16 TRs that have been processed by the EDMA on a given queue, read the event queue registers. This provides user/software visibility and is helpful for debugging real-time issues (typically post-mortem), involving multiple events and event sources.

The queue status register (QSTAT*n* EDMA_TPCC_QSTATN_i) includes fields for the start pointer EDMA_TPCC_QSTATN_i[3:0] STRTPTR which provides the offset to the head entry of an event. It also includes a field called EDMA_TPCC_QSTATN_i[12:8] NUMVAL that provides the total number of valid entries residing in the event queue at a given instance of time. The EDMA_TPCC_QSTATN_i[3:0] STRTPTR is used to index appropriately into the 16 event entries. EDMA_TPCC_QSTATN_i[12:8] NUMVAL number of entries starting from STRTPTR are indicative of events still queued in the respective queue. The remaining entry must be read to determine what's already de-queued and submitted to the associated transfer controller.

12.2.11.3 Queue Resource Tracking

The EDMA_TPCC event queue includes watermarking/threshold logic that allows to keep track of maximum usage of all event queues. This is useful for debugging real-time deadline violations that may result from head-of-line blocking on a given EDMA event queue.

The maximum number of events are programed that the queue up in an event queue by programming the threshold value (between 0 to 15) in the queue watermark threshold A register EDMA_TPCC_QWMTHRA. The maximum queue usage is recorded actively in the watermark EDMA_TPCC_QSTATN_i[20:16] WM field of the queue status register, that keeps getting updated based on a comparison of number of valid entries, which is also visible in the EDMA_TPCC_QSTATN_i[12:8] NUMVAL bit and the maximum number of entries.

If the queue usage is exceeded, this status is visible in the EDMA_TPCC registers: the QTHRXCD*n* bits in the channel controller error register EDMA_TPCC_CCERR[7:0] and the EDMA_TPCC_QSTATN_i[24] THRXCD bit, where *n* stands for the event queue number. Any bits that are set in EDMA_TPCC_CCERR also generate an EDMA_TPCC error interrupt.

12.2.11.4 Performance Considerations

The device system bus infrastructure arbitrates bus requests from all of the masters (TCs, CPU(S), and other bus masters) to the shared slave resources (peripherals and memories).

The priorities of transfer requests (read and write commands) from the EDMA transfer controllers with respect to other masters within the device IRQ_CROSSBAR are programmed using the Control Module registers. The EDMA_TPCC_QUEPRI register has no affect.

Therefore, the priority of unloading queues has a secondary affect compared to the priority of the transfers as they are executed by the EDMA_TPTC (dictated by the priority set using the Control Module registers, refer to *Control Module Register Manual* in *Control Module* chapter).

12.2.12 EDMA Transfer Controller (EDMA_TPTC)

The EDMA channel controller is the user-interface of the EDMA and the EDMA transfer controller (EDMA_TPTC) is the data movement engine of the EDMA controller. The EDMA_TPCC submits transfer requests (TR) to the EDMA_TPTC and the EDMA_TPTC performs the data transfers dictated by the TR, so the EDMA_TPTC is a slave to the EDMA_TPCC.

12.2.12.1 Architecture Details

12.2.12.1.1 Command Fragmentation

The TC read and write controllers in conjunction with the source and destination register sets are responsible for issuing optimally-sized reads and writes to the slave endpoints. An optimally-sized command is defined by the transfer controller default burst size (DBS), which is defined in *EDMA_TPTC Configuration*.

The EDMA_TPTC attempts to issue the largest possible command size as limited by the DBS value or the EDMA_TPCC_ABCNT_n[15:0] ACNT and EDMA_TPCC_ABCNT_n[31:16] BCNT value of the TR. EDMA_TPTC obeys the following rules:

- The read/write controllers always issue commands less than or equal to the DBS value.
- The first command of a 1D transfer command always aligns the address of subsequent commands to the DBS value.

Table 12-16 lists the TR segmentation rules that are followed by the EDMA_TPTC. In summary, if the EDMA_TPCC_ABCNT_n[15:0] ACNT value is larger than the DBS value, then the EDMA_TPTC breaks the EDMA_TPCC_ABCNT_n[15:0] ACNT array into DBS-sized commands to the source/destination addresses. Each EDMA_TPCC_ABCNT_n[31:16] BCNT number of arrays are then serviced in succession.

For BCNT arrays of ACNT bytes (that is, a 2D transfer), if the EDMA_TPCC_ABCNT_n[15:0] ACNT value is less than or equal to the DBS value, then the TR may be optimized into a 1D-transfer in order to maximize efficiency. The optimization takes place if the EDMA_TPTC recognizes that the 2D-transfer is organized as a single dimension (EDMA_TPCC_ABCNT_n[15:0] ACNT == EDMA_TPCC_BIDX_n) and the ACNT value is a power of 2.

Table 12-16 lists conditions in which the optimizations are performed.

Table 12-16. Read/Write Command Optimization Rules

ACNT ≤ DBS	ACNT is power of 2	BIDX = ACNT	BCNT ≤ 1023	SAM/DAM = Increment	Description
Yes	Yes	Yes	Yes	Yes	Optimized
No	x	x	x	x	Not Optimized
x	No	x	x	x	Not Optimized
x	x	No	x	x	Not Optimized
x	x	x	No	x	Not Optimized
x	x	x	x	No	Not Optimized

12.2.12.1.2 TR Pipelining

TR pipelining refers to the ability of the source active set to proceed ahead of the destination active set. Essentially, the reads for a given TR may already be in progress while the writes of a previous TR may not have completed.

The number of outstanding TRs is limited by the number of destination FIFO register entries.

TR pipelining is useful for maintaining throughput on back-to-back small TRs. It minimizes the startup overhead because reads start in the background of a previous TR writes.

Example 12-4. Command Fragmentation (DBS = 64)

The pseudo code:

1. EDMA_TPTCn_PCNT[15:0] ACNT = 8, EDMA_TPTCn_PCNT[31:16] BCNT = 8, EDMA_TPTCn_PBIDX[15:0] SBIDX = 8, EDMA_TPTCn_PBIDX[31:16] DBIDX = 10, EDMA_TPTCn_PSRC[31:0] SADDR = 64, EDMA_TPTCn_SADST[31:0] DADDR = 191

   Read Controller: This is optimized from a 2D-transfer to a 1D-transfer such that the read side is equivalent to EDMA_TPTCn_PCNT[15:0] ACNT = 64, EDMA_TPTCn_PCNT[31:16] BCNT = 1.

   Cmd0 = 64 byte

   Write Controller: Because DBIDX != ACNT, it is not optimized.

   Cmd0 = 8 byte, Cmd1 = 8 byte, Cmd2 = 8 byte, Cmd3 = 8 byte, Cmd4 = 8 byte, Cmd5 = 8 byte, Cmd6 = 8 byte, Cmd7 = 8 byte.
2. EDMA_TPTCn_PCNT[15:0] ACNT=128, EDMA_TPTCn_PCNT[31:16] BCNT = 1, EDMA_TPTCn_PSRC[31:0] SADDR = 63, EDMA_TPTCn_SADST[31:0] DADDR = 513

   Read Controller: Read address is not aligned.

   Cmd0 = 1 byte, (now the SADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 63 bytes

   Write Controller: The write address is also not aligned.

   Cmd0 = 63 bytes, (now the DADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 1 byte

12.2.12.1.3 Performance Tuning

By default, reads are as issued as fast as possible. In some cases, the reads issued by the EDMA_TPTC could fill the available command buffering for a slave, delaying other (potentially higher priority) masters from successfully submitting commands to that slave. The rate at which read commands are issued by the EDMA_TPTC is controlled by the EDMA_TPTCn_RDRATE register. The EDMA_TPTCn_RDRATE register defines the number of cycles that the EDMA_TPTC read controller waits before issuing subsequent commands for a given TR, thus minimizing the chance of the EDMA_TPTC consuming all available slave resources. The EDMA_TPTCn_RDRATE[2:0] RDRATE value must be set to a relatively small value if the transfer controller is targeted for high priority transfers and to a higher value if the transfer controller is targeted for low priority transfers.

In contrast, the Write Interface does not have any performance turning knobs because writes always have an interval between commands as write commands are submitted along with the associated write data.

12.2.12.2 Memory Protection

The transfer controller plays an important role in handling proxy memory protection. There are two access properties associated with a transfer: for instance, the privilege id (system-wide identification assigned to a master) of the master initiating the transfer, and the privilege level (user versus supervisor) used to program the transfer. This information is maintained in the PaRAM set when it is programmed in the channel controller. When a TR is submitted to the transfer controller, this information is made available to the EDMA_TPTC and used by the EDMA_TPTC while issuing read and write commands. The read or write commands have the same privilege identification, and privilege level as that programmed in the EDMA transfer in the channel controller.

12.2.12.3 Error Generation

Errors are generated if enabled under three conditions:

- EDMA_TPTC detection of an error signaled by the source or destination address.
- Attempt to read or write to an invalid address in the configuration memory map.

- Detection of a constant addressing mode TR violating the constant addressing mode transfer rules (the source/destination addresses and source/destination indexes must be aligned to 32 bytes).

Either or all error types may be disabled. If an error bit is set and enabled, the error interrupt for the concerned transfer controller is generated.

12.2.12.4 Debug Features

The DMA program register set, DMA source active register set, and the destination FIFO register set are used to derive a brief history of TRs serviced through the transfer controller.

Additionally, the EDMA_TPTC status register EDMA_TPTCn_TCSTAT has dedicated bit fields to indicate the ongoing activity within different parts of the transfer controller:

- The EDMA_TPTCn_TCSTAT[1] SRCACTV bit indicates whether the source active set is active.
- The EDMA_TPTCn_TCSTAT[6:4] DSTACTV bit indicates the number of TRs resident in the destination register active set at a given instance.
- The EDMA_TPTCn_TCSTAT[0] PROGBUSY bit indicates whether a valid TR is present in the DMA program set.

NOTE: If the TRs are in progression, it must realize that there is a chance that the values read from the EDMA_TPTC status registers will be inconsistent since the EDMA_TPTC changes the values of these registers due to ongoing activities.

It is recommended that to ensure no additional submission of TRs to the EDMA_TPTC in order to facilitate ease of debug.

12.2.12.4.1 Destination FIFO Register Pointer

The destination FIFO register pointer is implemented as a circular buffer with the start pointer being EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR and a buffer depth of usually 2 or 4. The EDMA_TPTC maintains two important status details in EDMA_TPTCn_TCSTAT that are used during advanced debugging, if necessary. The EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR is a start pointer, the index to the head of the destination FIFO register. The EDMA_TPTCn_TCSTAT[6:4] DSTACTV is a counter for the number of valid (occupied) entries. These registers are used to get a brief history of transfers.

Examples of some register field values and their interpretation:

- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x0 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x0 implies that no TRs are stored in the destination FIFO register.
- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x1 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 1 and the second pending TR is read from the destination FIFO register entry 2.
- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x3 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 3 and the second pending TR is read from the destination FIFO register entry 0.

12.2.13 Event Dataflow

This section summarizes the data flow of a single event, from the time the event is latched to the channel controller to the time the transfer completion code is returned. The following steps list the sequence of EDMA_TPCC activity:

1. Event is asserted from an external source (peripheral or external interrupt). This also is similar for a manually-triggered, chained-triggered, or QDMA-triggered event. The event is latched into the EDMA_TPCC_ER[31:0]E*n* / EDMA_TPCC_ERH[31:0] E*n* (or EDMA_TPCC_CER[31:0] E*n* / EDMA_TPCC_CERH[31:0] E*n*, EDMA_TPCC_ESR[31:0] E*n* / EDMA_TPCC_ESRH[31:0] E*n*, EDMA_TPCC_QER[7:0] E*n*) bit.
2. Once an event is prioritized and queued into the appropriate event queue, the EDMA_TPCC_SER[31:0] E*n* \ EDMA_TPCC_SERH[31:0] E*n* (or EDMA_TPCC_QSER[7:0] E*n*) bit is set to inform the event prioritization / processing logic to disregard this event since it is already in the

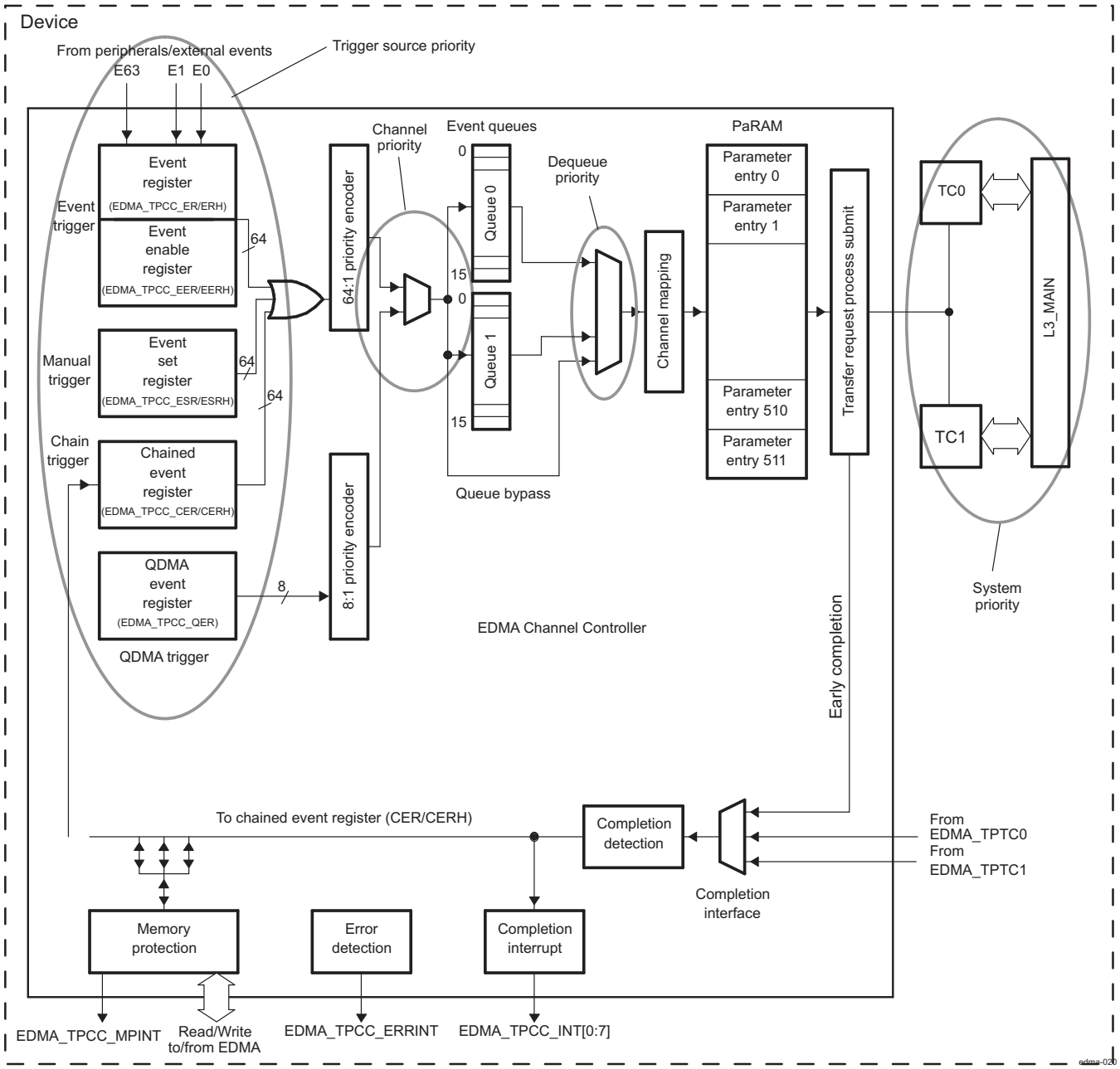
queue. Alternatively, if the transfer controller and the event queue are empty, then the event bypasses the queue.

3. The EDMA_TPCC processing and the submission logic evaluates the appropriate PaRAM set and determines whether it is a non-null and non-dummy transfer request (TR).
4. The EDMA_TPCC clears the EDMA_TPCC_ER[31:0] E*n*/ EDMA_TPCC_ERH[31:0] E*n* (or EDMA_TPCC_CER[31:0] E*n* / EDMA_TPCC_CERH[31:0] E*n*, EDMA_TPCC_ESR[31:0]E*n* / EDMA_TPCC_ESRH[31:0] E*n*, EDMA_TPCC_QER[31:0] E*n*) bit and the EDMA_TPCC_SER[31:0] E*n*/EDMA_TPCC_SERH[31:0] E*n* bit as soon as it determines the TR is non-null. In the case of a null set, the EDMA_TPCC_SER[31:0] E*n*/ EDMA_TPCC_SERH[31:0] E*n* bit remains set. It submits the non-null/non-dummy TR to the associated transfer controller. If the TR was programmed for early completion, the EDMA_TPCC immediately sets the interrupt pending register (EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC] - 32).
5. If the TR was programmed for normal completion, the EDMA_TPCC sets the interrupt pending register (EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC]) when the EDMA_TPTC informs the EDMA_TPCC about completion of the transfer (returns transfer completion codes).
6. The EDMA_TPCC programs the associated EDMA_TPTC's Program Register Set with the TR.
7. The TR is then passed to the Source Active set and the DST FIFO Register Set, if both the register sets are available.
8. The Read Controller processes the TR by issuing read commands to the source slave endpoint. The Read Data lands in the Data FIFO of the EDMA_TPTC*n*.
9. As soon as sufficient data is available, the Write Controller begins processing the TR by issuing write commands to the destination slave endpoint.
10. This continues until the TR completes and the EDMA_TPTC*n* then signals completion status to the EDMA_TPCC.

12.2.14 EDMA Controller Prioritization

The EDMA controller has many implementation rules to deal with concurrent events/channels, transfers, etc. The following subsections detail various arbitration details whenever there might be occurrence of concurrent activity. Figure 12-19 shows the different places EDMA priorities come into play.

Figure 12-19. EDMA Prioritization



12.2.14.1 Channel Priority

The EDMA event registers EDMA_TPCC_ER and EDMA_TPCC_ERH capture up to 64 events, the QDMA event register EDMA_TPCC_QER captures QDMA events for all QDMA channels therefore, it is possible for events to occur simultaneously on the DMA/QDMA event inputs. For events arriving simultaneously, the event associated with the lowest channel number is prioritized for submission to the event queues (for DMA events, channel 0 has the highest priority and channel 63 has the lowest priority, for QDMA events, channel 0 has the highest priority and channel 7 has the lowest priority). This mechanism only sorts simultaneous events for submission to the event queues.

If a DMA and QDMA event occurs simultaneously, the DMA event always has prioritization against the QDMA event for submission to the event queues.

12.2.14.2 Trigger Source Priority

If a EDMA channel is associated with more than one trigger source (event trigger, manual trigger, and chain trigger), and if multiple events are set simultaneously for the same channel (EDMA_TPCC_ER[31:0] E*n* = 1, EDMA_TPCC_ESR[31:0] E*n* = 1 , EDMA_TPCC_CER[31:0] E*n* = 1) , then the EDMA_TPCC always services these events in the following priority order: event trigger (via EDMA_TPCC_ER) is higher priority than chain trigger (via EDMA_TPCC_CER) and chain trigger is higher priority than manual trigger (via EDMA_TPCC_ESR).

This implies that if for channel 0, both EDMA_TPCC_ER[0] E0 = 1 and EDMA_TPCC_CER[0] E0 = 1 at the same time, then the EDMA_TPCC_ER[0] E0 event is always queued before the EDMA_TPCC_CER[0] E0 event.

12.2.14.3 Dequeue Priority

The priority of the associated transfer request (TR) is further mitigated by which event queue is being used for event submission (dictated by EDMA_TPCC_DMAQNUMN_k and EDMA_TPCC_QDMAQNUM). For submission of a TR to the transfer request, events need to be de-queued from the event queues. Queue 0 has the highest dequeue priority and queue 1 the lowest.

12.2.15 Emulation Considerations

During debug when using the emulator, the CPU(s) may be halted on an execute packet boundary for single-stepping, benchmarking, profiling, or other debug purposes. During an emulation halt, the EDMA channel controller and transfer controller operations continue. Events continue to be latched and processed and transfer requests continue to be submitted and serviced.

Since EDMA is involved in servicing multiple master and slave peripherals, it is not feasible to have an independent behavior of the EDMA for emulation halts. EDMA functionality would be coupled with the peripherals it is servicing, which might have different behavior during emulation halts.

12.3 EDMA Transfer Examples

The EDMA channel controller performs a variety of transfers depending on the parameter configuration. The following sections provide a description and PaRAM configuration for some typical use case scenarios.

12.3.1 Block Move Example

The most basic transfer performed by the EDMA is a block move. During device operation it is often necessary to transfer a block of data from one location to another, usually between on-chip and off-chip memory.

In this example, a section of data is to be copied from external memory to internal L2 SRAM as shown in Figure 12-20.

The source address for the transfer is set to the start of the data block in external memory, and the destination address is set to the start of the data block in L2. If the data block is less than 64K bytes, the PaRAM configuration shown in Figure 12-21 holds true with the synchronization type set to A-synchronized and indexes cleared to 0. If the amount of data is greater than 64K bytes, EDMA_TPCC_ABCNT_n[31:16] BCNT and the B-indexes need to be set appropriately with the synchronization type set to AB-synchronized. The EDMA_TPCC_OPT_n[3] STATIC bit is set to prevent linking.

This transfer example may also be set up using QDMA. For successive transfer submissions, of a similar nature, the number of cycles used to submit the transfer are fewer depending on the number of changing transfer parameters. The QDMA trigger word must be programed to be the highest numbered offset in the PaRAM set that undergoes change.

Figure 12-21 shows the parameters Block Move transfer.

Figure 12-20. Block Move Example

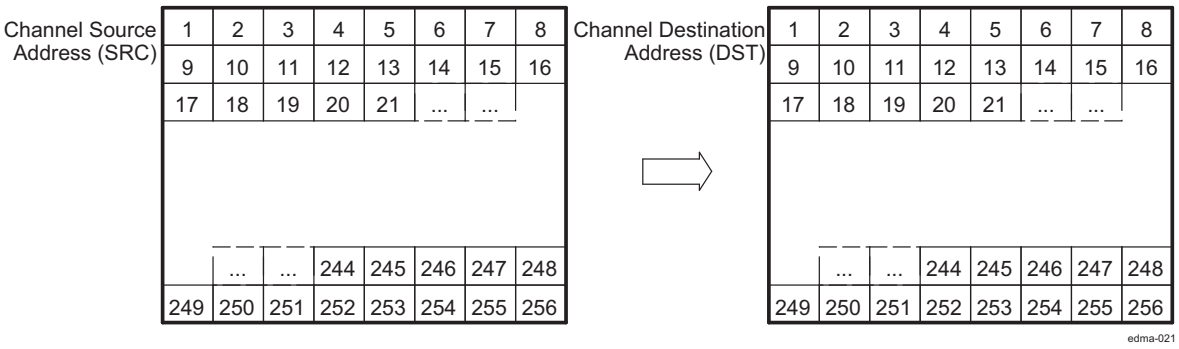


Figure 12-21. Block Move Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0010 0008h	
Channel Source Address (SRC)	
0001h	0100h
Channel Destination Address (DST)	
0000h	0000h
0000h	FFFFh
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b)Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[3] STATIC = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

12.3.2 Subframe Extraction Example

The EDMA can efficiently extract a small frame of data from a larger frame of data. By performing a 2D-to-1D transfer, the EDMA retrieves a portion of data for the CPU to process. In this example, a 640 × 480-pixel frame of video data is stored in external memory. Each pixel is represented by a 16-bit halfword. The CPU extracts a 16 × 12-pixel subframe of the image for processing. To facilitate more efficient processing time by the CPU, the EDMA places the subframe in internal L2 SRAM. Figure 12-22 shows the transfer of a subframe from external memory to L2.

The same PaRAM entry options are used for QDMA channels, as well as DMA channels. The EDMA_TPCC_OPT_n[3] STATIC bit is set to prevent linking. For successive transfers, only changed parameters need to be programmed before triggering the channel.

Figure 12-23 shows the parameters for Subframe Extraction transfer.

Figure 12-22. Subframe Extraction Transfer

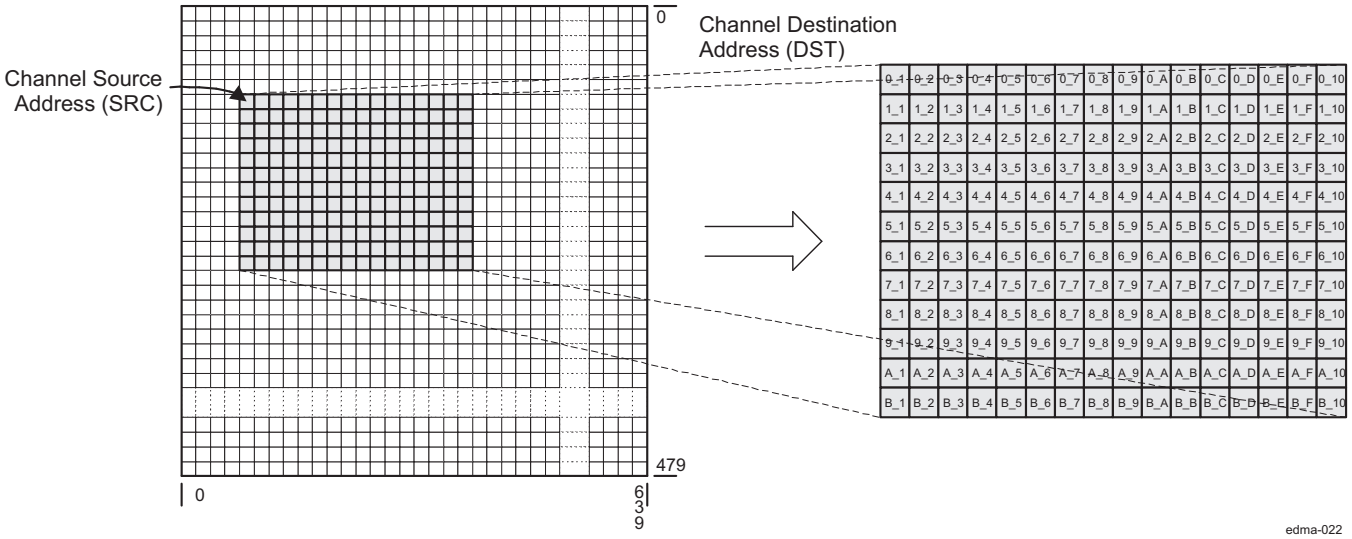


Figure 12-23. Subframe Extraction Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0010 000Ch	
Channel Source Address (SRC)	
000Ch	0020h
Channel Destination Address (DST)	
0020h	0500h
0000h	FFFFh
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b)Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[3] STATIC = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

12.3.3 Data Sorting Example

Many applications require the use of multiple data arrays, it is often desirable to have the arrays arranged such that the first elements of each array are adjacent, the second elements are adjacent, and so on. Often this is not how the data is presented to the device. Either data is transferred via a peripheral with the data arrays arriving one after the other or the arrays are located in memory with each array occupying a portion of contiguous memory spaces. For these instances, the EDMA can reorganize the data into the desired format.

To determine the parameter set values, the following need to be considered:

- ACNT - Program this to be the size in bytes of an element.
- BCNT - Program this to be the number of elements in a frame.
- CCNT - Program this to be the number of frames.
- SBIDX - Program this to be the size of the element or ACNT.
- DBIDX - CCNT × ACNT
- SCIDX - ACNT × BCNT
- DCIDX - ACNT

The synchronization type needs to be AB-synchronized and the EDMA_TPCC_OPT_n[3] STATIC bit is 0 to allow updates to the parameter set. It is advised to use normal EDMA channels for sorting.

It is not possible to sort this with a single trigger event. Instead, the channel can be programmed to be chained to itself. After BCNT elements get sorted, intermediate chaining could be used to trigger the channel again causing the transfer of the next BCNT elements and so on. Figure 12-25 shows the parameter set programming for this transfer, assuming channel 0 and an element size of 4 bytes.

Figure 12-24 shows the Data Sorting transfer

Figure 12-24. Data Sorting Example

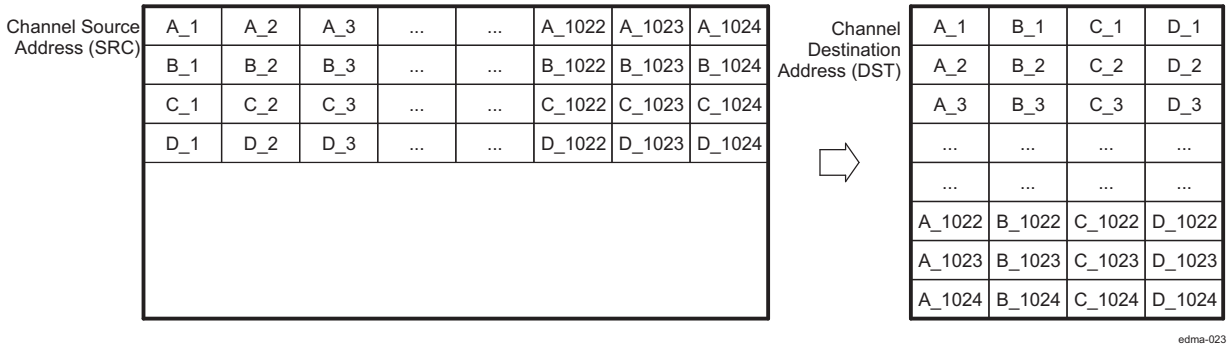


Figure 12-25. Data Sorting Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0090 0004h	
Channel Source Address (SRC)	
0400h	0004h
Channel Destination Address (DST)	
0010h	0001h
0000h	FFFFh
0001h	1000h
0000h	0004h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1
- EDMA_TPCC_OPT_n[23] ITCCHEN = 0x1

12.3.4 Setting Up an EDMA Transfer

The following list provides a quick guide for the typical steps involved in setting up a transfer.

Step 1. Initiating a DMA/QDMA channel

1. Determine the type of channel (QDMA or DMA) to be used.
2. Channel mapping
   i. If using a QDMA channel, program the EDMA_TPCC_QCHMAPN_j with the parameter set number to which the channel maps and the trigger word.
   ii. If using a DMA channel, program the EDMA_TPCC_DCHMAPN_m with the parameter set number to which the channel maps.
3. If the channel is being used in the context of a shadow region, ensure the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k for the region is properly set up to allow read write accesses to bits in the event registers and interrupt registers in the Shadow region memory map. The subsequent steps in this process should be done using the respective shadow region registers. (Shadow region descriptions and usage are provided in Section 12.2.7.1.)
4. Determine the type of triggering used.
   i. If external events are used for triggering (DMA channels), enable the respective event in EDMA_TPCC_EER / EDMA_TPCC_EERH by writing into EDMA_TPCC_EESR / EDMA_TPCC_EESRH.
   ii. If QDMA Channel is used, enable the channel in EDMA_TPCC_QEER by writing into EDMA_TPCC_QEESR.
5. Queue setup
   i. If a QDMA channel is used, set up the EDMA_TPCC_QDMAQNUM to map the channel to the respective event queue.
   ii. If a DMA channel is used, set up the EDMA_TPCC_DMAQNUMN_k to map the event to the respective event queue.

Step 2. Parameter set setup

1. Program the PaRAM set number associated with the channel. Note that

NOTE: If it is a QDMA channel, the PaRAM entry that is configured as trigger word is written to last. Alternatively, enable the QDMA channel (step 1-b-ii above) just before the write to the trigger word.

Step 3. Interrupt setup

1. Enable the interrupt in the EDMA_TPCC_IER / EDMA_TPCC_IERH by writing into EDMA_TPCC_IESR / EDMA_TPCC_IESRH.
2. Ensure that the EDMA_TPCC completion interrupt (either the global or the shadow region interrupt) is enabled properly in the device interrupt controller.
3. Ensure the EDMA_TPCC completion interrupt (this refers to either the Global interrupt or the shadow region interrupt) is enabled properly in the Device Interrupt controller.
4. Set up the interrupt controller properly to receive the expected EDMA interrupt.

Step 4. Initiate transfer

1. This step is highly dependent on the event trigger source:
   i. If the source is an external event coming from a peripheral, the peripheral will be enabled to start generating relevant EDMA events that can be latched to the EDMA_TPCC_ER transfer.
   ii. For QDMA events, writes to the trigger word (step 2-a above) will initiate the transfer.
   iii. Manually triggered transfers will be initiated by writes to the Event Set Registers EDMA_TPCC_ESR / EDMA_TPCC_ESRH.
   iv. Chained-trigger events initiate when a previous transfer returns a transfer completion code equal to the chained channel number.

Step 5. Wait for completion

1. If the interrupts are enabled as mentioned in step 3 above, then the EDMA_TPCC will generate a completion interrupt to the CPU whenever transfer completion results in setting the corresponding bits in the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH. The set bits must be cleared in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH by writing to corresponding bit in EDMA_TPCC_ICR / EDMA_TPCC_ICRH.
2. If polling for completion (interrupts not enabled in the device controller), then the application code can wait on the expected bits to be set in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH. Again, the set bits in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH must be manually cleared via EDMA_TPCC_ICR / EDMA_TPCC_ICRH before the next set of transfers is performed for the same transfer completion code values.

12.4 EDMA Debug Checklist and Programming Tips

This section lists some tips to keep in mind while debugging applications using the EDMA controller.

12.4.1 EDMA Debug Checklist

Table 12-17 provides some common issues and their probable causes and resolutions.

Table 12-17. Debug Checklist

Issue	Description/Solution
The transfer associated with the channel does not happen. The channel does not get serviced.	The EDMA_TPCC may not service a transfer request, even though the associated PaRAM set is programmed appropriately. Check for the following: 1) Verify that events are enabled, i.e., if an external/peripheral event is latched in Event Registers EDMA_TPCC_ER / EDMA_TPCC_ERH, check that the event is enabled in the Event Enable Registers EDMA_TPCC_EER / EDMA_TPCC_EERH. Similarly, for QDMA channels, check that QDMA events are appropriately enabled in the QDMA Event Enable Register EDMA_TPCC_QEER. 2) Verify that the DMA or QDMA Secondary Event Register EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER bits corresponding to the particular event or channel are not set.
The Secondary Event Registers bits are set, not allowing additional transfers to occur on a channel.	It is possible that a trigger event was received when the parameter set associated with the channel/event was a NULL set for a previous transfer on the channel. This is typical in two cases: 1) QDMA channels: Typically if the parameter set is non-static and expected to be terminated by a NULL set (i.e., EDMA_TPCC_OPT_n[3] STATIC = 0x0, EDMA_TPCC_LNK_n[15:0] LINK = 0xFFFF), the parameter set is updated with a NULL set after submission of the last TR. Because QDMA channels are auto-triggered, this update caused the generation of an event. An event generated for a NULL set causes an error condition and results in setting the bits corresponding to the QDMA channel in the EDMA_TPCC_QEMR and EDMA_TPCC_QSER. This will disable further prioritization of the channel. 2) DMA channels used in a continuous mode: The peripheral may be set up to continuously generate infinite events (for instance, in case of McASP, every time the data shifts out from the DXR register, it generates an XEVT). The parameter set may be programmed to expect only a finite number of events and to be terminated by a NULL link. After the expected number of events, the parameter set is reloaded with a NULL parameter set. Because the peripheral will generate additional events, an error condition is set in the EDMA_TPCC_SER[31:0] E*n* and EDMA_TPCC_EMR[31:0] E*n* set, preventing further event prioritization. Check the number of events received is limited to the expected number of events for which the parameter set is programmed, or check the bits corresponding to particular channel or event are not set in the Secondary event registers (EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER) and Event Missed Registers (EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR) before trying to perform subsequent transfers for the event/channel.
Completion interrupts are not asserted, or no further interrupts are received after the first completion interrupt.	Check the following: 1) The interrupt generation is enabled in the EDMA_TPCC_OPT_n of the associated PaRAM set (EDMA_TPCC_OPT_n[20] TCINTEN = 0x1 and/or EDMA_TPCC_OPT_n[20] ITCINTEN = 0x1). 2) The interrupts are enabled in the EDMA Channel Controller, via the Interrupt Enable Registers (EDMA_TPCC_IER / EDMA_TPCC_IERH). 3) The corresponding interrupts are enabled in the device interrupt controller. 4) The set interrupts are cleared in the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH) before exiting the transfer completion interrupt service routine (ISR). See Section 12.2.9.1.2 *Clearing Transfer Complerion Interrupts* for details on writing EDMA ISRs. 5) If working with shadow region interrupts, make sure that the DMA Region Access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k) are set up properly, because the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers act as secondary enables for shadow region completion interrupts, along with the EDMA_TPCC_IER / EDMA_TPCC_IERH registers. If working with shadow region interrupts, make sure that the bits corresponding to the transfer completion code EDMA_TPCC_OPT_n[17:12] TCC value are also enabled in the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers. For instance, if the PaRAM set associated with Channel 0 returns a completion code of 63 EDMA_TPCC_OPT_n[17:12] TCC = 63, ensure that EDMA_TPCC_DRAEHM_k[31] E63 is also set for a shadow region completion interrupt because the interrupt pending register bit set will be EDMA_TPCC_IPRH[31] I63 (not EDMA_TPCC_IPR[0] I0).

12.4.2 EDMA Programming Tips

1. For several registers, the setting and clearing of bits needs to be done via separate dedicated registers.

   For example, the Event Register (EDMA_TPCC_ER / EDMA_TPCC_ERH) can only be cleared by writing a 1 to the corresponding bits in the Event Clear Registers (EDMA_TPCC_ECR / EDMA_TPCC_ECRH). Similarly, the Event Enable Register (EDMA_TPCC_EER / EDMA_TPCC_EERH) bits can only be set with writing of 0x1 to the Event Enable Set Registers (EDMA_TPCC_EESR / EDMA_TPCC_EESRH) and cleared with writing of 0x1 to the corresponding bits in the Event Enable Clear Register (EDMA_TPCC_EECR / EDMA_TPCC_EECRH).
2. Writes to the shadow region memory maps are governed by region access registers (EDMA_TPCC_DRAE / EDMA_TPCC_DRAEHM_k / EDMA_TPCC_QRAEN_k). If the appropriate channels are not enabled in these registers, read/write access to the shadow region memory map is not enabled.
3. When working with shadow region completion interrupts, ensure that the DMA Region Access Registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k) for every region are set in a mutually exclusive way (unless it is a requirement for an application). If there is an overlap in the allocated channels and transfer completion codes (setting of Interrupt Pending Register bits) in the region resource allocation, it results in multiple shadow region completion interrupts.

   For example, if EDMA_TPCC_DRAEM_k.DRAEM_0[0] E0 and EDMA_TPCC_DRAEM_k.DRAEM_1[0] E0 are both set, then on completion of a transfer that returns a TCC = 0x0, they will generate both shadow region 0 and 1 completion interrupts.
4. While programming a non-dummy parameter set, ensure the EDMA_TPCC_CCNT_n[15:0] CCNT is not left to zero.
5. Enable the EDMA_TPCC error interrupt in the device controller and attach an interrupt service routine (ISR) to ensure that error conditions are not missed in an application and are appropriately addressed with the ISR.
6. Depending on the application, it can want to break large transfers into smaller transfers and use self-chaining to prevent starvation of other events in an event queue.
7. In applications where a large transfer is broken into sets of small transfers using chaining or other methods, it chooses to use the early chaining option to reduce the time between the sets of transfers and increase the throughput.

   However, keep in mind that with early completion, all data might have not been received at the end point when completion is reported because the EDMA_TPCC internally signals completion when the TR is submitted to the EDMA_TPTC, potentially before any data has been transferred.
8. The event queue entries can be observed to determine the last few events if there is a system failure (provided the entries were not bypassed).

12.5 EDMA Request Map

For the EDMA request map, refer to Table 2-14 for the 16xx/18xx, Table 3-14 for the 68xx, and Table 1-12 for the 14xx.

12.6 EDMA Register Manual

12.6.1 EDMA Registers

12.6.1.1 EDMA_TPCC Registers

Table 12-18 lists the memory-mapped registers for the EDMA_TPCC. All register offset addresses not listed in Table 12-18 should be considered as reserved locations and the register contents should not be modified.

Table 12-18. EDMA_TPCC Registers

Offset	Acronym	Register Name	Section
0h	EDMA_TPCC_PID	EDMA_TPCC_PID	Section 12.6.1.1.1
4h	EDMA_TPCC_CCCFG	EDMA_TPCC_CCCFG	Section 12.6.1.1.2
200h	EDMA_TPCC_QCHMAPN	EDMA_TPCC_QCHMAPN	Section 12.6.1.1.3
240h	EDMA_TPCC_DMAQNUMN	EDMA_TPCC_DMAQNUMN	Section 12.6.1.1.4
260h	EDMA_TPCC_QDMAQNUM	EDMA_TPCC_QDMAQNUM	Section 12.6.1.1.5
280h	EDMA_TPCC_QUETCMAP	EDMA_TPCC_QUETCMAP	Section 12.6.1.1.6
284h	EDMA_TPCC_QUEPRI	EDMA_TPCC_QUEPRI	Section 12.6.1.1.7
300h	EDMA_TPCC_EMR	EDMA_TPCC_EMR	Section 12.6.1.1.8
304h	EDMA_TPCC_EMRH	EDMA_TPCC_EMRH	Section 12.6.1.1.9
308h	EDMA_TPCC_EMCR	EDMA_TPCC_EMCR	Section 12.6.1.1.10
30Ch	EDMA_TPCC_EMCRH	EDMA_TPCC_EMCRH	Section 12.6.1.1.11
310h	EDMA_TPCC_QEMR	EDMA_TPCC_QEMR	Section 12.6.1.1.12
314h	EDMA_TPCC_QEMCR	EDMA_TPCC_QEMCR	Section 12.6.1.1.13
318h	EDMA_TPCC_CCERR	EDMA_TPCC_CCERR	Section 12.6.1.1.14
31Ch	EDMA_TPCC_CCERRCLR	EDMA_TPCC_CCERRCLR	Section 12.6.1.1.15
320h	EDMA_TPCC_EEVAL	EDMA_TPCC_EEVAL	Section 12.6.1.1.16
340h	EDMA_TPCC_DRAEN	EDMA_TPCC_DRAEN	Section 12.6.1.1.17
340h	EDMA_TPCC_DRAEM	EDMA_TPCC_DRAEM	Section 12.6.1.1.18
344h	EDMA_TPCC_DRAEHM	EDMA_TPCC_DRAEHM	Section 12.6.1.1.19
380h	EDMA_TPCC_QRAEN	EDMA_TPCC_QRAEN	Section 12.6.1.1.20
400h	EDMA_TPCC_QNEM	EDMA_TPCC_QNEM	Section 12.6.1.1.21
400h	EDMA_TPCC_QNE0	EDMA_TPCC_QNE0	Section 12.6.1.1.22
404h	EDMA_TPCC_QNE1	EDMA_TPCC_QNE1	Section 12.6.1.1.23
408h	EDMA_TPCC_QNE2	EDMA_TPCC_QNE2	Section 12.6.1.1.24
40Ch	EDMA_TPCC_QNE3	EDMA_TPCC_QNE3	Section 12.6.1.1.25
410h	EDMA_TPCC_QNE4	EDMA_TPCC_QNE4	Section 12.6.1.1.26
414h	EDMA_TPCC_QNE5	EDMA_TPCC_QNE5	Section 12.6.1.1.27
418h	EDMA_TPCC_QNE6	EDMA_TPCC_QNE6	Section 12.6.1.1.28
41Ch	EDMA_TPCC_QNE7	EDMA_TPCC_QNE7	Section 12.6.1.1.29
420h	EDMA_TPCC_QNE8	EDMA_TPCC_QNE8	Section 12.6.1.1.30
424h	EDMA_TPCC_QNE9	EDMA_TPCC_QNE9	Section 12.6.1.1.31
428h	EDMA_TPCC_QNE10	EDMA_TPCC_QNE10	Section 12.6.1.1.32
42Ch	EDMA_TPCC_QNE11	EDMA_TPCC_QNE11	Section 12.6.1.1.33
430h	EDMA_TPCC_QNE12	EDMA_TPCC_QNE12	Section 12.6.1.1.34
434h	EDMA_TPCC_QNE13	EDMA_TPCC_QNE13	Section 12.6.1.1.35
438h	EDMA_TPCC_QNE14	EDMA_TPCC_QNE14	Section 12.6.1.1.36
43Ch	EDMA_TPCC_QNE15	EDMA_TPCC_QNE15	Section 12.6.1.1.37
600h	EDMA_TPCC_QSTATN	EDMA_TPCC_QSTATN	Section 12.6.1.1.38
620h	EDMA_TPCC_QWMTHRA	EDMA_TPCC_QWMTHRA	Section 12.6.1.1.39
640h	EDMA_TPCC_CCSTAT	EDMA_TPCC_CCSTAT	Section 12.6.1.1.40
700h	EDMA_TPCC_AETCTL	EDMA_TPCC_AETCTL	Section 12.6.1.1.41
704h	EDMA_TPCC_AETSTAT	EDMA_TPCC_AETSTAT	Section 12.6.1.1.42
708h	EDMA_TPCC_AETCMD	EDMA_TPCC_AETCMD	Section 12.6.1.1.43

Table 12-18. EDMA_TPCC Registers (continued)

Offset	Acronym	Register Name	Section
1000h	EDMA_TPCC_ER	EDMA_TPCC_ER	Section 12.6.1.1.44
1004h	EDMA_TPCC_ERH	EDMA_TPCC_ERH	Section 12.6.1.1.45
1008h	EDMA_TPCC_ECR	EDMA_TPCC_ECR	Section 12.6.1.1.46
100Ch	EDMA_TPCC_ECRH	EDMA_TPCC_ECRH	Section 12.6.1.1.47
1010h	EDMA_TPCC_ESR	EDMA_TPCC_ESR	Section 12.6.1.1.48
1014h	EDMA_TPCC_ESRH	EDMA_TPCC_ESRH	Section 12.6.1.1.49
1018h	EDMA_TPCC_CER	EDMA_TPCC_CER	Section 12.6.1.1.50
101Ch	EDMA_TPCC_CERH	EDMA_TPCC_CERH	Section 12.6.1.1.51
1020h	EDMA_TPCC_EER	EDMA_TPCC_EER	Section 12.6.1.1.52
1024h	EDMA_TPCC_EERH	EDMA_TPCC_EERH	Section 12.6.1.1.53
1028h	EDMA_TPCC_EECR	EDMA_TPCC_EECR	Section 12.6.1.1.54
102Ch	EDMA_TPCC_EECRH	EDMA_TPCC_EECRH	Section 12.6.1.1.55
1030h	EDMA_TPCC_EESR	EDMA_TPCC_EESR	Section 12.6.1.1.56
1034h	EDMA_TPCC_EESRH	EDMA_TPCC_EESRH	Section 12.6.1.1.57
1038h	EDMA_TPCC_SER	EDMA_TPCC_SER	Section 12.6.1.1.58
103Ch	EDMA_TPCC_SERH	EDMA_TPCC_SERH	Section 12.6.1.1.59
1040h	EDMA_TPCC_SECR	EDMA_TPCC_SECR	Section 12.6.1.1.60
1044h	EDMA_TPCC_SECRH	EDMA_TPCC_SECRH	Section 12.6.1.1.61
1050h	EDMA_TPCC_IER	EDMA_TPCC_IER	Section 12.6.1.1.62
1054h	EDMA_TPCC_IERH	EDMA_TPCC_IERH	Section 12.6.1.1.63
1058h	EDMA_TPCC_IECR	EDMA_TPCC_IECR	Section 12.6.1.1.64
105Ch	EDMA_TPCC_IECRH	EDMA_TPCC_IECRH	Section 12.6.1.1.65
1060h	EDMA_TPCC_IESR	EDMA_TPCC_IESR	Section 12.6.1.1.66
1064h	EDMA_TPCC_IESRH	EDMA_TPCC_IESRH	Section 12.6.1.1.67
1068h	EDMA_TPCC_IPR	EDMA_TPCC_IPR	Section 12.6.1.1.68
106Ch	EDMA_TPCC_IPRH	EDMA_TPCC_IPRH	Section 12.6.1.1.69
1070h	EDMA_TPCC_ICR	EDMA_TPCC_ICR	Section 12.6.1.1.70
1074h	EDMA_TPCC_ICRH	EDMA_TPCC_ICRH	Section 12.6.1.1.71
1078h	EDMA_TPCC_IEVAL	EDMA_TPCC_IEVAL	Section 12.6.1.1.72
1080h	EDMA_TPCC_QER	EDMA_TPCC_QER	Section 12.6.1.1.73
1084h	EDMA_TPCC_QEER	EDMA_TPCC_QEER	Section 12.6.1.1.74
1088h	EDMA_TPCC_QEECR	EDMA_TPCC_QEECR	Section 12.6.1.1.75
108Ch	EDMA_TPCC_QEESR	EDMA_TPCC_QEESR	Section 12.6.1.1.76
1090h	EDMA_TPCC_QSER	EDMA_TPCC_QSER	Section 12.6.1.1.77
1094h	EDMA_TPCC_QSECR	EDMA_TPCC_QSECR	Section 12.6.1.1.78
2000h	EDMA_TPCC_SHADOW_N	EDMA_TPCC_SHADOW_N	Section 12.6.1.1.79
2000h	EDMA_TPCC_ER_RN	EDMA_TPCC_ER_RN	Section 12.6.1.1.80
2004h	EDMA_TPCC_ERH_RN	EDMA_TPCC_ERH_RN	Section 12.6.1.1.81
2008h	EDMA_TPCC_ECR_RN	EDMA_TPCC_ECR_RN	Section 12.6.1.1.82
200Ch	EDMA_TPCC_ECRH_RN	EDMA_TPCC_ECRH_RN	Section 12.6.1.1.83
2010h	EDMA_TPCC_ESR_RN	EDMA_TPCC_ESR_RN	Section 12.6.1.1.84
2014h	EDMA_TPCC_ESRH_RN	EDMA_TPCC_ESRH_RN	Section 12.6.1.1.85
2018h	EDMA_TPCC_CER_RN	EDMA_TPCC_CER_RN	Section 12.6.1.1.86
201Ch	EDMA_TPCC_CERH_RN	EDMA_TPCC_CERH_RN	Section 12.6.1.1.87
2020h	EDMA_TPCC_EER_RN	EDMA_TPCC_EER_RN	Section 12.6.1.1.88
2024h	EDMA_TPCC_EERH_RN	EDMA_TPCC_EERH_RN	Section 12.6.1.1.89
2028h	EDMA_TPCC_EECR_RN	EDMA_TPCC_EECR_RN	Section 12.6.1.1.90

Table 12-18. EDMA_TPCC Registers (continued)

Offset	Acronym	Register Name	Section
202Ch	EDMA_TPCC_EECRH_RN	EDMA_TPCC_EECRH_RN	Section 12.6.1.1.91
2030h	EDMA_TPCC_EESR_RN	EDMA_TPCC_EESR_RN	Section 12.6.1.1.92
2034h	EDMA_TPCC_EESRH_RN	EDMA_TPCC_EESRH_RN	Section 12.6.1.1.93
2038h	EDMA_TPCC_SER_RN	EDMA_TPCC_SER_RN	Section 12.6.1.1.94
203Ch	EDMA_TPCC_SERH_RN	EDMA_TPCC_SERH_RN	Section 12.6.1.1.95
2040h	EDMA_TPCC_SECR_RN	EDMA_TPCC_SECR_RN	Section 12.6.1.1.96
2044h	EDMA_TPCC_SECRH_RN	EDMA_TPCC_SECRH_RN	Section 12.6.1.1.97
2050h	EDMA_TPCC_IER_RN	EDMA_TPCC_IER_RN	Section 12.6.1.1.98
2054h	EDMA_TPCC_IERH_RN	EDMA_TPCC_IERH_RN	Section 12.6.1.1.99
2058h	EDMA_TPCC_IECR_RN	EDMA_TPCC_IECR_RN	Section 12.6.1.1.100
205Ch	EDMA_TPCC_IECRH_RN	EDMA_TPCC_IECRH_RN	Section 12.6.1.1.101
2060h	EDMA_TPCC_IESR_RN	EDMA_TPCC_IESR_RN	Section 12.6.1.1.102
2064h	EDMA_TPCC_IESRH_RN	EDMA_TPCC_IESRH_RN	Section 12.6.1.1.103
2068h	EDMA_TPCC_IPR_RN	EDMA_TPCC_IPR_RN	Section 12.6.1.1.104
206Ch	EDMA_TPCC_IPRH_RN	EDMA_TPCC_IPRH_RN	Section 12.6.1.1.105
2070h	EDMA_TPCC_ICR_RN	EDMA_TPCC_ICR_RN	Section 12.6.1.1.106
2074h	EDMA_TPCC_ICRH_RN	EDMA_TPCC_ICRH_RN	Section 12.6.1.1.107
2078h	EDMA_TPCC_IEVAL_RN	EDMA_TPCC_IEVAL_RN	Section 12.6.1.1.108
2080h	EDMA_TPCC_QER_RN	EDMA_TPCC_QER_RN	Section 12.6.1.1.109
2084h	EDMA_TPCC_QEER_RN	EDMA_TPCC_QEER_RN	Section 12.6.1.1.110
2088h	EDMA_TPCC_QEECR_RN	EDMA_TPCC_QEECR_RN	Section 12.6.1.1.111
208Ch	EDMA_TPCC_QEESR_RN	EDMA_TPCC_QEESR_RN	Section 12.6.1.1.112
2090h	EDMA_TPCC_QSER_RN	EDMA_TPCC_QSER_RN	Section 12.6.1.1.113
2094h	EDMA_TPCC_QSECR_RN	EDMA_TPCC_QSECR_RN	Section 12.6.1.1.114
4000h	EDMA_TPCC_PARAMSET	EDMA_TPCC_PARAMSET	Section 12.6.1.1.115
4000h	EDMA_TPCC_OPT	EDMA_TPCC_OPT	Section 12.6.1.1.116
4004h	EDMA_TPCC_SRC	EDMA_TPCC_SRC	Section 12.6.1.1.117
4008h	EDMA_TPCC_ABCNT	EDMA_TPCC_ABCNT	Section 12.6.1.1.118
400Ch	EDMA_TPCC_DST	EDMA_TPCC_DST	Section 12.6.1.1.119
4010h	EDMA_TPCC_BIDX	EDMA_TPCC_BIDX	Section 12.6.1.1.120
4014h	EDMA_TPCC_LNK	EDMA_TPCC_LNK	Section 12.6.1.1.121
4018h	EDMA_TPCC_CIDX	EDMA_TPCC_CIDX	Section 12.6.1.1.122
401Ch	EDMA_TPCC_CCNT	EDMA_TPCC_CCNT	Section 12.6.1.1.123

12.6.1.1.1 EDMA_TPCC_PID Register (Offset = 0h) [reset = 4001AB00h]

EDMA_TPCC_PID is shown in Figure 12-26 and described in Table 12-19.

Return to Summary Table.

Peripheral ID Register

Figure 12-26. EDMA_TPCC_PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RES1		FUNC											
R-1h		R-0h		R-1h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-15h					R-3h			R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-19. EDMA_TPCC_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Scheme: Used to distinguish between old ID scheme and current. Spare bit to encode future schemes EDMA uses 'new scheme' indicated with value of 0x1.
29-28	RES1	R	0h	RESERVE FIELD
27-16	FUNC	R	1h	Function indicates a software compatible module family.
15-11	RTL	R	15h	RTL Version
10-8	MAJOR	R	3h	Major Revision
7-6	CUSTOM	R	0h	Custom revision field: Not used on this version of EDMA.
5-0	MINOR	R	0h	Minor Revision

12.6.1.1.2 EDMA_TPCC_CCCFG Register (Offset = 4h) [reset = X]

EDMA_TPCC_CCCFG is shown in Figure 12-27 and described in Table 12-20.

Return to Summary Table.

CC Configuration Register

Figure 12-27. EDMA_TPCC_CCCFG Register

31	30	29	28	27	26	25	24
RES2						MPEXIST	CHMAPEXIST
R-x						R-0h	R-0h
23	22	21	20	19	18	17	16
RES3		NUMREGN		RES4	NUM_EVQUE		
R-0h		R-2h		R-x	R-1h		
15	14	13	12	11	10	9	8
RES5	NUMPAENTRY			RES6	NUMINTCH		
R-x	R-3h			R-x	R-4h		
7	6	5	4	3	2	1	0
RES7	NUMQDMACH			RES8	NUMDMACH		
R-x	R-4h			R-x	R-5h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset; -x = value is indeterminate after reset; -y = device-specific

Table 12-20. EDMA_TPCC_CCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RES2	R	x	Reserved
25	MPEXIST	R	0h	Memory Protection Existence MPEXIST = 0 : No memory protection. MPEXIST = 1 : Memory Protection logic included.
24	CHMAPEXIST	R	0h	Channel Mapping Existence CHMAPEXIST = 0 : No Channel mapping. CHMAPEXIST = 1 : Channel mapping logic included.
23-22	RES3	R	0h	Reserved
21-20	NUMREGN	R	2h	Number of MP and Shadow regions NUMREGN = 0 : 0 Regions NUMREGN = 1 : 2 Regions NUMREGN = 2 : 4 Regions NUMREGN = 3 : 8 Regions
19	RES4	R	x	Reserved
18-16	NUM_EVQUE	R	1h	Number of Queues/Number of TCs NUMTC = 0 : 1 TC/Event Queue NUMTC = 1 : 2 TC/Event Queues NUMTC = 2 : 3 TC/Event Queues ... NUMTC = 7 : 8 TC/Event Queues
15	RES5	R	x	Reserved

Table 12-20. EDMA_TPCC_CCCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-12	NUMPAENTRY	R	3h	Number of PaRAM entries NUMPAENTRY = 0 : 16 entries NUMPAENTRY = 1 : 32 entries (unsupported setting) NUMPAENTRY = 2 : 64 entries NUMPAENTRY = 3 : 128 entries NUMPAENTRY = 4 : 256 entries NUMPAENTRY = 5 : 512 entries Others : reserved
11	RES6	R	x	Reserved
10-8	NUMINTCH	R	4h	Number of Interrupt Channels NUMINTCH = 0 : reserved NUMINTCH = 1 : 8 Interrupt channels NUMINTCH = 2 : 16 Interrupt channels NUMINTCH = 3 : 32 Interrupt channels NUMINTCH = 4 : 64 Interrupt channels Others : reserved
7	RES7	R	x	Reserved
6-4	NUMQDMACH	R	4h	Number of QDMA Channels NUMQDMACH = 0 : No QDMA Channels NUMQDMACH = 1 : 2 QDMA Channels NUMQDMACH = 2 : 4 QDMA Channels NUMQDMACH = 3 : 6 QDMA Channels NUMQDMACH = 4 : 8 QDMA Channels Others : reserved
3	RES8	R	x	Reserved
2-0	NUMDMACH	R	5h	Number of DMA Channels NUMDMACH = 0 : No DMA Channels NUMDMACH = 1 : 4 DMA Channels NUMDMACH = 2 : 8 DMA Channels NUMDMACH = 3 : 16 DMA Channels NUMDMACH = 4 : 32 DMA Channels NUMDMACH = 5 : 64 DMA Channels Others : reserved

12.6.1.1.3 EDMA_TPCC_QCHMAPN Register (Offset = 200h) [reset = 0h]

EDMA_TPCC_QCHMAPN is shown in Figure 12-28 and described in Table 12-21.

Return to Summary Table.

QDMA Channel N Mapping Register

Figure 12-28. EDMA_TPCC_QCHMAPN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES10															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES10		PAENTRY									TRWORD			RESERVED	
R-0h		R/W-0h									R/W-0h			R-	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-21. EDMA_TPCC_QCHMAPN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES10	R	0h	RESERVE FIELD
13-5	PAENTRY	R/W	0h	PaRAM Entry number for QDMA Channel N.
4-2	TRWORD	R/W	0h	TRWORD points to the specific trigger word of the PaRAM Entry defined by PAENTRY. A write to the trigger word results in a QDMA Event being recognized.
1-0	RESERVED	R		

12.6.1.1.4 EDMA_TPCC_DMAQNUMN Register (Offset = 240h) [reset = 0h]

EDMA_TPCC_DMAQNUMN is shown in Figure 12-29 and described in Table 12-22.

Return to Summary Table.

DMA Queue Number Register n Contains the Event queue number to be used for the corresponding DMA Channel.

Figure 12-29. EDMA_TPCC_DMAQNUMN Register

31	30	29	28	27	26	25	24
RES11	E7			RES12	E6		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RES13	E5			RES14	E4		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RES15	E3			RES16	E2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RES17	E1			RES18	E0		
R-0h	R/W-0h			R-0h	R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-22. EDMA_TPCC_DMAQNUMN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RES11	R	0h	RESERVE FIELD
30-28	E7	R/W	0h	DMA Queue Number for event #7
27	RES12	R	0h	RESERVE FIELD
26-24	E6	R/W	0h	DMA Queue Number for event #6
23	RES13	R	0h	RESERVE FIELD
22-20	E5	R/W	0h	DMA Queue Number for event #5
19	RES14	R	0h	RESERVE FIELD
18-16	E4	R/W	0h	DMA Queue Number for event #4
15	RES15	R	0h	RESERVE FIELD
14-12	E3	R/W	0h	DMA Queue Number for event #3
11	RES16	R	0h	RESERVE FIELD
10-8	E2	R/W	0h	DMA Queue Number for event #2
7	RES17	R	0h	RESERVE FIELD
6-4	E1	R/W	0h	DMA Queue Number for event #1
3	RES18	R	0h	RESERVE FIELD
2-0	E0	R/W	0h	DMA Queue Number for event #0

12.6.1.1.5 EDMA_TPCC_QDMAQNUM Register (Offset = 260h) [reset = 0h]

EDMA_TPCC_QDMAQNUM is shown in Figure 12-30 and described in Table 12-23.

Return to Summary Table.

QDMA Queue Number Register Contains the Event queue number to be used for the corresponding QDMA Channel.

Figure 12-30. EDMA_TPCC_QDMAQNUM Register

31	30	29	28	27	26	25	24
RES19	E7			RES20	E6		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RES21	E5			RES22	E4		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RES23	E3			RES24	E2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RES25	E1			RES26	E0		
R-0h	R/W-0h			R-0h	R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-23. EDMA_TPCC_QDMAQNUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RES19	R	0h	RESERVE FIELD
30-28	E7	R/W	0h	QDMA Queue Number for event #7
27	RES20	R	0h	RESERVE FIELD
26-24	E6	R/W	0h	QDMA Queue Number for event #6
23	RES21	R	0h	RESERVE FIELD
22-20	E5	R/W	0h	QDMA Queue Number for event #5
19	RES22	R	0h	RESERVE FIELD
18-16	E4	R/W	0h	QDMA Queue Number for event #4
15	RES23	R	0h	RESERVE FIELD
14-12	E3	R/W	0h	QDMA Queue Number for event #3
11	RES24	R	0h	RESERVE FIELD
10-8	E2	R/W	0h	QDMA Queue Number for event #2
7	RES25	R	0h	RESERVE FIELD
6-4	E1	R/W	0h	QDMA Queue Number for event #1
3	RES26	R	0h	RESERVE FIELD
2-0	E0	R/W	0h	QDMA Queue Number for event #0

12.6.1.1.6 EDMA_TPCC_QUETCMAP Register (Offset = 280h) [reset = 10h]

EDMA_TPCC_QUETCMAP is shown in Figure 12-31 and described in Table 12-24.

Return to Summary Table.

Queue to TC Mapping

Figure 12-31. EDMA_TPCC_QUETCMAP Register

31	30	29	28	27	26	25	24
RES27							
R-0h							
23	22	21	20	19	18	17	16
RES27							
R-0h							
15	14	13	12	11	10	9	8
RES27							
R-0h							
7	6	5	4	3	2	1	0
RES27	TCNUMQ1			RES28	TCNUMQ0		
R-0h	R/W-1h			R-0h	R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-24. EDMA_TPCC_QUETCMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RES27	R	0h	RESERVE FIELD
6-4	TCNUMQ1	R/W	1h	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.
3	RES28	R	0h	RESERVE FIELD
2-0	TCNUMQ0	R/W	0h	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.

12.6.1.1.7 EDMA_TPCC_QUEPRI Register (Offset = 284h) [reset = 0h]

EDMA_TPCC_QUEPRI is shown in Figure 12-32 and described in Table 12-25.

Return to Summary Table.

Queue Priority

Figure 12-32. EDMA_TPCC_QUEPRI Register

31	30	29	28	27	26	25	24
RES29							
R-0h							
23	22	21	20	19	18	17	16
RES29							
R-0h							
15	14	13	12	11	10	9	8
RES29							
R-0h							
7	6	5	4	3	2	1	0
RES29	PRIQ1			RES30	PRIQ0		
R-0h	R/W-0h			R-0h	R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-25. EDMA_TPCC_QUEPRI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RES29	R	0h	RESERVE FIELD
6-4	PRIQ1	R/W	0h	Priority Level for Queue 1 Dictates the priority level used for the OPTIONS field programmation for Qn TRs. Sets the priority used for TC read and write commands.
3	RES30	R	0h	RESERVE FIELD
2-0	PRIQ0	R/W	0h	Priority Level for Queue 0 Dictates the priority level used for the OPTIONS field programmation for Qn TRs. Sets the priority used for TC read and write commands.

12.6.1.1.8 EDMA_TPCC_EMR Register (Offset = 300h) [reset = 0h]

EDMA_TPCC_EMR is shown in Figure 12-33 and described in Table 12-26.

Return to Summary Table.

Event Missed Register: The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER) Set Events (ESR) and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including QEMR/CCERR) were previously clear) then an error will be signaled with TPCC error interrupt.

Figure 12-33. EDMA_TPCC_EMR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-26. EDMA_TPCC_EMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event Missed #31
30	E30	R	0h	Event Missed #30
29	E29	R	0h	Event Missed #29
28	E28	R	0h	Event Missed #28
27	E27	R	0h	Event Missed #27
26	E26	R	0h	Event Missed #26
25	E25	R	0h	Event Missed #25
24	E24	R	0h	Event Missed #24
23	E23	R	0h	Event Missed #23
22	E22	R	0h	Event Missed #22
21	E21	R	0h	Event Missed #21
20	E20	R	0h	Event Missed #20
19	E19	R	0h	Event Missed #19
18	E18	R	0h	Event Missed #18
17	E17	R	0h	Event Missed #17
16	E16	R	0h	Event Missed #16
15	E15	R	0h	Event Missed #15
14	E14	R	0h	Event Missed #14
13	E13	R	0h	Event Missed #13
12	E12	R	0h	Event Missed #12
11	E11	R	0h	Event Missed #11
10	E10	R	0h	Event Missed #10
9	E9	R	0h	Event Missed #9
8	E8	R	0h	Event Missed #8
7	E7	R	0h	Event Missed #7
6	E6	R	0h	Event Missed #6

Table 12-26. EDMA_TPCC_EMR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	E5	R	0h	Event Missed #5
4	E4	R	0h	Event Missed #4
3	E3	R	0h	Event Missed #3
2	E2	R	0h	Event Missed #2
1	E1	R	0h	Event Missed #1
0	E0	R	0h	Event Missed #0

12.6.1.1.9 EDMA_TPCC_EMRH Register (Offset = 304h) [reset = 0h]

EDMA_TPCC_EMRH is shown in Figure 12-34 and described in Table 12-27.

Return to Summary Table.

Event Missed Register (High Part): The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER) Set Events (ESR) and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including QEMR/CCERR) were previously clear) then an error will be signaled with TPCC error interrupt.

Figure 12-34. EDMA_TPCC_EMRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-27. EDMA_TPCC_EMRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event Missed #63
30	E62	R	0h	Event Missed #62
29	E61	R	0h	Event Missed #61
28	E60	R	0h	Event Missed #60
27	E59	R	0h	Event Missed #59
26	E58	R	0h	Event Missed #58
25	E57	R	0h	Event Missed #57
24	E56	R	0h	Event Missed #56
23	E55	R	0h	Event Missed #55
22	E54	R	0h	Event Missed #54
21	E53	R	0h	Event Missed #53
20	E52	R	0h	Event Missed #52
19	E51	R	0h	Event Missed #51
18	E50	R	0h	Event Missed #50
17	E49	R	0h	Event Missed #49
16	E48	R	0h	Event Missed #48
15	E47	R	0h	Event Missed #47
14	E46	R	0h	Event Missed #46
13	E45	R	0h	Event Missed #45
12	E44	R	0h	Event Missed #44
11	E43	R	0h	Event Missed #43
10	E42	R	0h	Event Missed #42
9	E41	R	0h	Event Missed #41
8	E40	R	0h	Event Missed #40
7	E39	R	0h	Event Missed #39
6	E38	R	0h	Event Missed #38

Table 12-27. EDMA_TPCC_EMRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	E37	R	0h	Event Missed #37
4	E36	R	0h	Event Missed #36
3	E35	R	0h	Event Missed #35
2	E34	R	0h	Event Missed #34
1	E33	R	0h	Event Missed #33
0	E32	R	0h	Event Missed #32

12.6.1.1.10 EDMA_TPCC_EMCR Register (Offset = 308h) [reset = 0h]

EDMA_TPCC_EMCR is shown in Figure 12-35 and described in Table 12-28.

Return to Summary Table.

Event Missed Clear Register: CPU write of '1' to the EMCR.En bit causes the EMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.

Figure 12-35. EDMA_TPCC_EMCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-28. EDMA_TPCC_EMCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event Missed Clear #31
30	E30	W	0h	Event Missed Clear #30
29	E29	W	0h	Event Missed Clear #29
28	E28	W	0h	Event Missed Clear #28
27	E27	W	0h	Event Missed Clear #27
26	E26	W	0h	Event Missed Clear #26
25	E25	W	0h	Event Missed Clear #25
24	E24	W	0h	Event Missed Clear #24
23	E23	W	0h	Event Missed Clear #23
22	E22	W	0h	Event Missed Clear #22
21	E21	W	0h	Event Missed Clear #21
20	E20	W	0h	Event Missed Clear #20
19	E19	W	0h	Event Missed Clear #19
18	E18	W	0h	Event Missed Clear #18
17	E17	W	0h	Event Missed Clear #17
16	E16	W	0h	Event Missed Clear #16
15	E15	W	0h	Event Missed Clear #15
14	E14	W	0h	Event Missed Clear #14
13	E13	W	0h	Event Missed Clear #13
12	E12	W	0h	Event Missed Clear #12
11	E11	W	0h	Event Missed Clear #11
10	E10	W	0h	Event Missed Clear #10
9	E9	W	0h	Event Missed Clear #9
8	E8	W	0h	Event Missed Clear #8
7	E7	W	0h	Event Missed Clear #7
6	E6	W	0h	Event Missed Clear #6
5	E5	W	0h	Event Missed Clear #5

Table 12-28. EDMA_TPCC_EMCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E4	W	0h	Event Missed Clear #4
3	E3	W	0h	Event Missed Clear #3
2	E2	W	0h	Event Missed Clear #2
1	E1	W	0h	Event Missed Clear #1
0	E0	W	0h	Event Missed Clear #0

12.6.1.1.11 EDMA_TPCC_EMCRH Register (Offset = 30Ch) [reset = 0h]

EDMA_TPCC_EMCRH is shown in Figure 12-36 and described in Table 12-29.

Return to Summary Table.

Event Missed Clear Register (High Part): CPU write of '1' to the EMCR.En bit causes the EMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.

Figure 12-36. EDMA_TPCC_EMCRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-29. EDMA_TPCC_EMCRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event Missed Clear #63
30	E62	W	0h	Event Missed Clear #62
29	E61	W	0h	Event Missed Clear #61
28	E60	W	0h	Event Missed Clear #60
27	E59	W	0h	Event Missed Clear #59
26	E58	W	0h	Event Missed Clear #58
25	E57	W	0h	Event Missed Clear #57
24	E56	W	0h	Event Missed Clear #56
23	E55	W	0h	Event Missed Clear #55
22	E54	W	0h	Event Missed Clear #54
21	E53	W	0h	Event Missed Clear #53
20	E52	W	0h	Event Missed Clear #52
19	E51	W	0h	Event Missed Clear #51
18	E50	W	0h	Event Missed Clear #50
17	E49	W	0h	Event Missed Clear #49
16	E48	W	0h	Event Missed Clear #48
15	E47	W	0h	Event Missed Clear #47
14	E46	W	0h	Event Missed Clear #46
13	E45	W	0h	Event Missed Clear #45
12	E44	W	0h	Event Missed Clear #44
11	E43	W	0h	Event Missed Clear #43
10	E42	W	0h	Event Missed Clear #42
9	E41	W	0h	Event Missed Clear #41
8	E40	W	0h	Event Missed Clear #40
7	E39	W	0h	Event Missed Clear #39
6	E38	W	0h	Event Missed Clear #38
5	E37	W	0h	Event Missed Clear #37

Table 12-29. EDMA_TPCC_EMCRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E36	W	0h	Event Missed Clear #36
3	E35	W	0h	Event Missed Clear #35
2	E34	W	0h	Event Missed Clear #34
1	E33	W	0h	Event Missed Clear #33
0	E32	W	0h	Event Missed Clear #32

12.6.1.1.12 EDMA_TPCC_QEMR Register (Offset = 310h) [reset = 0h]

EDMA_TPCC_QEMR is shown in Figure 12-37 and described in Table 12-30.

Return to Summary Table.

QDMA Event Missed Register: The QDMA Event Missed register is set if 2 QDMA events are detected without the first event being cleared or if a Null TR is serviced.. If any bit in the QEMR register is set (and all errors (including EMR/CCERR) were previously clear) then an error will be signaled with TPCC error interrupt.

Figure 12-37. EDMA_TPCC_QEMR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES31															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES31								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-30. EDMA_TPCC_QEMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES31	R	0h	RESERVE FIELD
7	E7	R	0h	Event Missed #7
6	E6	R	0h	Event Missed #6
5	E5	R	0h	Event Missed #5
4	E4	R	0h	Event Missed #4
3	E3	R	0h	Event Missed #3
2	E2	R	0h	Event Missed #2
1	E1	R	0h	Event Missed #1
0	E0	R	0h	Event Missed #0

12.6.1.1.13 EDMA_TPCC_QEMCR Register (Offset = 314h) [reset = 0h]

EDMA_TPCC_QEMCR is shown in Figure 12-38 and described in Table 12-31.

Return to Summary Table.

QDMA Event Missed Clear Register: CPU write of '1' to the QEMCR.En bit causes the QEMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.

Figure 12-38. EDMA_TPCC_QEMCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES32															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES32								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-31. EDMA_TPCC_QEMCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES32	R	0h	RESERVE FIELD
7	E7	W	0h	Event Missed Clear #7
6	E6	W	0h	Event Missed Clear #6
5	E5	W	0h	Event Missed Clear #5
4	E4	W	0h	Event Missed Clear #4
3	E3	W	0h	Event Missed Clear #3
2	E2	W	0h	Event Missed Clear #2
1	E1	W	0h	Event Missed Clear #1
0	E0	W	0h	Event Missed Clear #0

12.6.1.1.14 EDMA_TPCC_CCERR Register (Offset = 318h) [reset = 0h]

EDMA_TPCC_CCERR is shown in Figure 12-39 and described in Table 12-32.

Return to Summary Table.

CC Error Register

Figure 12-39. EDMA_TPCC_CCERR Register

31	30	29	28	27	26	25	24
RES33							
R-0h							
23	22	21	20	19	18	17	16
RES33							TCERR
R-0h							R-0h
15	14	13	12	11	10	9	8
RES34							
R-0h							
7	6	5	4	3	2	1	0
QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-32. EDMA_TPCC_CCERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RES33	R	0h	RESERVE FIELD
16	TCERR	R	0h	Transfer Completion Code Error: TCCERR = 0 : Total number of allowed TCCs outstanding has not been reached. TCCERR = 1 : Total number of allowed TCCs has been reached. TCCERR can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors were previously clear) then an error will be signaled with TPCC error interrupt.
15-8	RES34	R	0h	RESERVE FIELD
7	QTHRXCD7	R	0h	Queue Threshold Error for Q7: QTHRXCD7 = 0 : Watermark/threshold has not been exceeded. QTHRXCD7 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD7 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
6	QTHRXCD6	R	0h	Queue Threshold Error for Q6: QTHRXCD6 = 0 : Watermark/threshold has not been exceeded. QTHRXCD6 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD6 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
5	QTHRXCD5	R	0h	Queue Threshold Error for Q5: QTHRXCD5 = 0 : Watermark/threshold has not been exceeded. QTHRXCD5 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD5 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.

Table 12-32. EDMA_TPCC_CCERR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	QTHRXCD4	R	0h	Queue Threshold Error for Q4: QTHRXCD4 = 0 : Watermark/threshold has not been exceeded. QTHRXCD4 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD4 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
3	QTHRXCD3	R	0h	Queue Threshold Error for Q3: QTHRXCD3 = 0 : Watermark/threshold has not been exceeded. QTHRXCD3 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD3 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
2	QTHRXCD2	R	0h	Queue Threshold Error for Q2: QTHRXCD2 = 0 : Watermark/threshold has not been exceeded. QTHRXCD2 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD2 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
1	QTHRXCD1	R	0h	Queue Threshold Error for Q1: QTHRXCD1 = 0 : Watermark/threshold has not been exceeded. QTHRXCD1 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD1 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
0	QTHRXCD0	R	0h	Queue Threshold Error for Q0: QTHRXCD0 = 0 : Watermark/threshold has not been exceeded. QTHRXCD0 = 1 : Watermark/threshold has been exceeded. CCERR.QTHRXCD0 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.

12.6.1.1.15 EDMA_TPCC_CCERRCLR Register (Offset = 31Ch) [reset = 0h]

EDMA_TPCC_CCERRCLR is shown in Figure 12-40 and described in Table 12-33.

Return to Summary Table.

CC Error Clear Register

Figure 12-40. EDMA_TPCC_CCERRCLR Register

31	30	29	28	27	26	25	24
RES35							
R-0h							
23	22	21	20	19	18	17	16
RES35							TCERR
R-0h							W-0h
15	14	13	12	11	10	9	8
RES36							
R-0h							
7	6	5	4	3	2	1	0
QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-33. EDMA_TPCC_CCERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RES35	R	0h	RESERVE FIELD
16	TCERR	W	0h	Clear Error for CCERR.TCERR: Write of '1' clears the value of CCERR bit N. Writes of '0' have no affect.
15-8	RES36	R	0h	RESERVE FIELD
7	QTHRXCD7	W	0h	Clear error for CCERR.QTHRXCD7: Write of '1' clears the values of QSTAT7.WM QSTAT7.THRXCD CCERR.QTHRXCD7 Writes of '0' have no affect.
6	QTHRXCD6	W	0h	Clear error for CCERR.QTHRXCD6: Write of '1' clears the values of QSTAT6.WM QSTAT6.THRXCD CCERR.QTHRXCD6 Writes of '0' have no affect.
5	QTHRXCD5	W	0h	Clear error for CCERR.QTHRXCD5: Write of '1' clears the values of QSTAT5.WM QSTAT5.THRXCD CCERR.QTHRXCD5 Writes of '0' have no affect.
4	QTHRXCD4	W	0h	Clear error for CCERR.QTHRXCD4: Write of '1' clears the values of QSTAT4.WM QSTAT4.THRXCD CCERR.QTHRXCD4 Writes of '0' have no affect.
3	QTHRXCD3	W	0h	Clear error for CCERR.QTHRXCD3: Write of '1' clears the values of QSTAT3.WM QSTAT3.THRXCD CCERR.QTHRXCD3 Writes of '0' have no affect.
2	QTHRXCD2	W	0h	Clear error for CCERR.QTHRXCD2: Write of '1' clears the values of QSTAT2.WM QSTAT2.THRXCD CCERR.QTHRXCD2 Writes of '0' have no affect.
1	QTHRXCD1	W	0h	Clear error for CCERR.QTHRXCD1: Write of '1' clears the values of QSTAT1.WM QSTAT1.THRXCD CCERR.QTHRXCD1 Writes of '0' have no affect.
0	QTHRXCD0	W	0h	Clear error for CCERR.QTHRXCD0: Write of '1' clears the values of QSTAT0.WM QSTAT0.THRXCD CCERR.QTHRXCD0 Writes of '0' have no affect.

12.6.1.1.16 EDMA_TPCC_EEVAL Register (Offset = 320h) [reset = 0h]

EDMA_TPCC_EEVAL is shown in Figure 12-41 and described in Table 12-34.

Return to Summary Table.

Error Eval Register

Figure 12-41. EDMA_TPCC_EEVAL Register

31	30	29	28	27	26	25	24
RES37							
R-0h							
23	22	21	20	19	18	17	16
RES37							
R-0h							
15	14	13	12	11	10	9	8
RES37							
R-0h							
7	6	5	4	3	2	1	0
RES37						SET	EVAL
R-0h						W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-34. EDMA_TPCC_EEVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES37	R	0h	RESERVE FIELD
1	SET	W	0h	Error Interrupt Set: CPU write of '1' to the SET bit causes the TPCC error interrupt to be pulsed regardless of state of EMR/EMRH QEMR or CCERR. CPU write of '0' has no effect.
0	EVAL	W	0h	Error Interrupt Evaluate: CPU write of '1' to the EVAL bit causes the TPCC error interrupt to be pulsed if any errors have not been cleared in the EMR/EMRH QEMR or CCERR registers. CPU write of '0' has no effect.

12.6.1.1.17 EDMA_TPCC_DRAEN Register (Offset = 340h)

EDMA_TPCC_DRAEN is shown in Figure 12-42 and described in Table 12-35.

Return to Summary Table.

Bundle description is not available

Figure 12-42. EDMA_TPCC_DRAEN Register

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-35. EDMA_TPCC_DRAEN Register Field Descriptions

Bit	Field	Type	Reset	Description

12.6.1.1.18 EDMA_TPCC_DRAEM Register (Offset = 340h) [reset = 0h]

EDMA_TPCC_DRAEM is shown in Figure 12-43 and described in Table 12-36.

Return to Summary Table.

DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.

Figure 12-43. EDMA_TPCC_DRAEM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-36. EDMA_TPCC_DRAEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R/W	0h	DMA Region Access enable for Region M bit #31
30	E30	R/W	0h	DMA Region Access enable for Region M bit #30
29	E29	R/W	0h	DMA Region Access enable for Region M bit #29
28	E28	R/W	0h	DMA Region Access enable for Region M bit #28
27	E27	R/W	0h	DMA Region Access enable for Region M bit #27
26	E26	R/W	0h	DMA Region Access enable for Region M bit #26
25	E25	R/W	0h	DMA Region Access enable for Region M bit #25
24	E24	R/W	0h	DMA Region Access enable for Region M bit #24
23	E23	R/W	0h	DMA Region Access enable for Region M bit #23
22	E22	R/W	0h	DMA Region Access enable for Region M bit #22
21	E21	R/W	0h	DMA Region Access enable for Region M bit #21
20	E20	R/W	0h	DMA Region Access enable for Region M bit #20
19	E19	R/W	0h	DMA Region Access enable for Region M bit #19
18	E18	R/W	0h	DMA Region Access enable for Region M bit #18
17	E17	R/W	0h	DMA Region Access enable for Region M bit #17
16	E16	R/W	0h	DMA Region Access enable for Region M bit #16
15	E15	R/W	0h	DMA Region Access enable for Region M bit #15
14	E14	R/W	0h	DMA Region Access enable for Region M bit #14
13	E13	R/W	0h	DMA Region Access enable for Region M bit #13
12	E12	R/W	0h	DMA Region Access enable for Region M bit #12
11	E11	R/W	0h	DMA Region Access enable for Region M bit #11
10	E10	R/W	0h	DMA Region Access enable for Region M bit #10
9	E9	R/W	0h	DMA Region Access enable for Region M bit #9

Table 12-36. EDMA_TPCC_DRAEM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	E8	R/W	0h	DMA Region Access enable for Region M bit #8
7	E7	R/W	0h	DMA Region Access enable for Region M bit #7
6	E6	R/W	0h	DMA Region Access enable for Region M bit #6
5	E5	R/W	0h	DMA Region Access enable for Region M bit #5
4	E4	R/W	0h	DMA Region Access enable for Region M bit #4
3	E3	R/W	0h	DMA Region Access enable for Region M bit #3
2	E2	R/W	0h	DMA Region Access enable for Region M bit #2
1	E1	R/W	0h	DMA Region Access enable for Region M bit #1
0	E0	R/W	0h	DMA Region Access enable for Region M bit #0

12.6.1.1.19 EDMA_TPCC_DRAEHM Register (Offset = 344h) [reset = 0h]

EDMA_TPCC_DRAEHM is shown in Figure 12-44 and described in Table 12-37.

Return to Summary Table.

DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt. En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.

Figure 12-44. EDMA_TPCC_DRAEHM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-37. EDMA_TPCC_DRAEHM Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R/W	0h	DMA Region Access enable for Region M bit #63
30	E62	R/W	0h	DMA Region Access enable for Region M bit #62
29	E61	R/W	0h	DMA Region Access enable for Region M bit #61
28	E60	R/W	0h	DMA Region Access enable for Region M bit #60
27	E59	R/W	0h	DMA Region Access enable for Region M bit #59
26	E58	R/W	0h	DMA Region Access enable for Region M bit #58
25	E57	R/W	0h	DMA Region Access enable for Region M bit #57
24	E56	R/W	0h	DMA Region Access enable for Region M bit #56
23	E55	R/W	0h	DMA Region Access enable for Region M bit #55
22	E54	R/W	0h	DMA Region Access enable for Region M bit #54
21	E53	R/W	0h	DMA Region Access enable for Region M bit #53
20	E52	R/W	0h	DMA Region Access enable for Region M bit #52
19	E51	R/W	0h	DMA Region Access enable for Region M bit #51
18	E50	R/W	0h	DMA Region Access enable for Region M bit #50
17	E49	R/W	0h	DMA Region Access enable for Region M bit #49
16	E48	R/W	0h	DMA Region Access enable for Region M bit #48
15	E47	R/W	0h	DMA Region Access enable for Region M bit #47
14	E46	R/W	0h	DMA Region Access enable for Region M bit #46
13	E45	R/W	0h	DMA Region Access enable for Region M bit #45
12	E44	R/W	0h	DMA Region Access enable for Region M bit #44

Table 12-37. EDMA_TPCC_DRAEHM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	E43	R/W	0h	DMA Region Access enable for Region M bit #43
10	E42	R/W	0h	DMA Region Access enable for Region M bit #42
9	E41	R/W	0h	DMA Region Access enable for Region M bit #41
8	E40	R/W	0h	DMA Region Access enable for Region M bit #40
7	E39	R/W	0h	DMA Region Access enable for Region M bit #39
6	E38	R/W	0h	DMA Region Access enable for Region M bit #38
5	E37	R/W	0h	DMA Region Access enable for Region M bit #37
4	E36	R/W	0h	DMA Region Access enable for Region M bit #36
3	E35	R/W	0h	DMA Region Access enable for Region M bit #35
2	E34	R/W	0h	DMA Region Access enable for Region M bit #34
1	E33	R/W	0h	DMA Region Access enable for Region M bit #33
0	E32	R/W	0h	DMA Region Access enable for Region M bit #32

12.6.1.1.20 EDMA_TPCC_QRAEN Register (Offset = 380h) [reset = 0h]

EDMA_TPCC_QRAEN is shown in Figure 12-45 and described in Table 12-38.

Return to Summary Table.

QDMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any QDMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any QDMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region n interrupt.

Figure 12-45. EDMA_TPCC_QRAEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES38															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES38								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-38. EDMA_TPCC_QRAEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES38	R	0h	RESERVE FIELD
7	E7	R/W	0h	QDMA Region Access enable for Region M bit #7
6	E6	R/W	0h	QDMA Region Access enable for Region M bit #6
5	E5	R/W	0h	QDMA Region Access enable for Region M bit #5
4	E4	R/W	0h	QDMA Region Access enable for Region M bit #4
3	E3	R/W	0h	QDMA Region Access enable for Region M bit #3
2	E2	R/W	0h	QDMA Region Access enable for Region M bit #2
1	E1	R/W	0h	QDMA Region Access enable for Region M bit #1
0	E0	R/W	0h	QDMA Region Access enable for Region M bit #0

12.6.1.1.21 EDMA_TPCC_QNEM Register (Offset = 400h)

EDMA_TPCC_QNEM is shown in Figure 12-46 and described in Table 12-39.

Return to Summary Table.

Bundle description is not available

Figure 12-46. EDMA_TPCC_QNEM Register

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-39. EDMA_TPCC_QNEM Register Field Descriptions

Bit	Field	Type	Reset	Description

12.6.1.1.22 EDMA_TPCC_QNE0 Register (Offset = 400h) [reset = 0h]

EDMA_TPCC_QNE0 is shown in Figure 12-47 and described in Table 12-40.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 0

Figure 12-47. EDMA_TPCC_QNE0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES39															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES39								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-40. EDMA_TPCC_QNE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES39	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.23 EDMA_TPCC_QNE1 Register (Offset = 404h) [reset = 0h]

EDMA_TPCC_QNE1 is shown in Figure 12-48 and described in Table 12-41.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 1

Figure 12-48. EDMA_TPCC_QNE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES40															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES40								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-41. EDMA_TPCC_QNE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES40	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.24 EDMA_TPCC_QNE2 Register (Offset = 408h) [reset = 0h]

EDMA_TPCC_QNE2 is shown in Figure 12-49 and described in Table 12-42.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 2

Figure 12-49. EDMA_TPCC_QNE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES41															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES41								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-42. EDMA_TPCC_QNE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES41	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.25 EDMA_TPCC_QNE3 Register (Offset = 40Ch) [reset = 0h]

EDMA_TPCC_QNE3 is shown in Figure 12-50 and described in Table 12-43.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 3

Figure 12-50. EDMA_TPCC_QNE3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES42															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES42								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-43. EDMA_TPCC_QNE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES42	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.26 EDMA_TPCC_QNE4 Register (Offset = 410h) [reset = 0h]

EDMA_TPCC_QNE4 is shown in Figure 12-51 and described in Table 12-44.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 4

Figure 12-51. EDMA_TPCC_QNE4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES43															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES43								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-44. EDMA_TPCC_QNE4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES43	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.27 EDMA_TPCC_QNE5 Register (Offset = 414h) [reset = 0h]

EDMA_TPCC_QNE5 is shown in Figure 12-52 and described in Table 12-45.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 5

Figure 12-52. EDMA_TPCC_QNE5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES44															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES44								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-45. EDMA_TPCC_QNE5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES44	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.28 EDMA_TPCC_QNE6 Register (Offset = 418h) [reset = 0h]

EDMA_TPCC_QNE6 is shown in Figure 12-53 and described in Table 12-46.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 6

Figure 12-53. EDMA_TPCC_QNE6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES45															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES45								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-46. EDMA_TPCC_QNE6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES45	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.29 EDMA_TPCC_QNE7 Register (Offset = 41Ch) [reset = 0h]

EDMA_TPCC_QNE7 is shown in Figure 12-54 and described in Table 12-47.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 7

Figure 12-54. EDMA_TPCC_QNE7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES46															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES46								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-47. EDMA_TPCC_QNE7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES46	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.30 EDMA_TPCC_QNE8 Register (Offset = 420h) [reset = 0h]

EDMA_TPCC_QNE8 is shown in Figure 12-55 and described in Table 12-48.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 8

Figure 12-55. EDMA_TPCC_QNE8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES47															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES47								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-48. EDMA_TPCC_QNE8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES47	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.31 EDMA_TPCC_QNE9 Register (Offset = 424h) [reset = 0h]

EDMA_TPCC_QNE9 is shown in Figure 12-56 and described in Table 12-49.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 9

Figure 12-56. EDMA_TPCC_QNE9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES48															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES48								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-49. EDMA_TPCC_QNE9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES48	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.32 EDMA_TPCC_QNE10 Register (Offset = 428h) [reset = 0h]

EDMA_TPCC_QNE10 is shown in Figure 12-57 and described in Table 12-50.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 0

Figure 12-57. EDMA_TPCC_QNE10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES49															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES49								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-50. EDMA_TPCC_QNE10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES49	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.33 EDMA_TPCC_QNE11 Register (Offset = 42Ch) [reset = 0h]

EDMA_TPCC_QNE11 is shown in Figure 12-58 and described in Table 12-51.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 11

Figure 12-58. EDMA_TPCC_QNE11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES50															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES50								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-51. EDMA_TPCC_QNE11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES50	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.34 EDMA_TPCC_QNE12 Register (Offset = 430h) [reset = 0h]

EDMA_TPCC_QNE12 is shown in Figure 12-59 and described in Table 12-52.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 12

Figure 12-59. EDMA_TPCC_QNE12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES51															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES51								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-52. EDMA_TPCC_QNE12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES51	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.35 EDMA_TPCC_QNE13 Register (Offset = 434h) [reset = 0h]

EDMA_TPCC_QNE13 is shown in Figure 12-60 and described in Table 12-53.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 13

Figure 12-60. EDMA_TPCC_QNE13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES52															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES52								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-53. EDMA_TPCC_QNE13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES52	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.36 EDMA_TPCC_QNE14 Register (Offset = 438h) [reset = 0h]

EDMA_TPCC_QNE14 is shown in Figure 12-61 and described in Table 12-54.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 14

Figure 12-61. EDMA_TPCC_QNE14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES53															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES53								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-54. EDMA_TPCC_QNE14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES53	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.37 EDMA_TPCC_QNE15 Register (Offset = 43Ch) [reset = 0h]

EDMA_TPCC_QNE15 is shown in Figure 12-62 and described in Table 12-55.

Return to Summary Table.

Event Queue Entry Diagram for Queue n - Entry 15

Figure 12-62. EDMA_TPCC_QNE15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES54															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES54								ETYPE		ENUM					
R-0h								R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-55. EDMA_TPCC_QNE15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES54	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

12.6.1.1.38 EDMA_TPCC_QSTATN Register (Offset = 600h) [reset = 0h]

EDMA_TPCC_QSTATN is shown in Figure 12-63 and described in Table 12-56.

Return to Summary Table.

QSTATn Register Set

Figure 12-63. EDMA_TPCC_QSTATN Register

31	30	29	28	27	26	25	24
RES55							THRXCD
R-0h							R-0h
23	22	21	20	19	18	17	16
RES56			WM				
R-0h			R-0h				
15	14	13	12	11	10	9	8
RES57			NUMVAL				
R-0h			R-0h				
7	6	5	4	3	2	1	0
RES58				STRTPTR			
R-0h				R-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-56. EDMA_TPCC_QSTATN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RES55	R	0h	RESERVE FIELD
24	THRXCD	R	0h	Threshold Exceeded: THRXCD = 0 : Threshold specified by QWMTHR(A B).Qn has not been exceeded. THRXCD = 1 : Threshold specified by QWMTHR(A B).Qn has been exceeded. QSTATn.THRXCD is cleared via CCERR.WMCLRn bit.
23-21	RES56	R	0h	RESERVE FIELD
20-16	WM	R	0h	Watermark for Maximum Queue Usage: Watermark tracks the most entries that have been in QueueN since reset or since the last time that the watermark (WM) was cleared. QSTATn.WM is cleared via CCERR.WMCLRn bit. Legal values = 0x0 (empty) to 0x10 (full)
15-13	RES57	R	0h	RESERVE FIELD
12-8	NUMVAL	R	0h	Number of Valid Entries in QueueN: Represents the total number of entries residing in the Queue Manager FIFO at a given instant. Always enabled. Legal values = 0x0 (empty) to 0x10 (full)
7-4	RES58	R	0h	RESERVE FIELD
3-0	STRTPTR	R	0h	Start Pointer: Represents the offset to the head entry of QueueN in units of entries. Always enabled. Legal values = 0x0 (0th entry) to 0xF (15th entry)

12.6.1.1.39 EDMA_TPCC_QWMTHRA Register (Offset = 620h) [reset = 1010h]

EDMA_TPCC_QWMTHRA is shown in Figure 12-64 and described in Table 12-57.

Return to Summary Table.

Queue Threshold A for Q[3:0]: CCERR.QTHRXCDn and QSTATn.THRXCD error bit is set when the number of Events in QueueN at an instant in time (visible via QSTATn.NUMVAL) equals or exceeds the value specified by QWMTHRA.Qn. Legal values = 0x0 (ever used?) to 0x10 (ever full?) A value of 0x11 disables threshold errors.

Figure 12-64. EDMA_TPCC_QWMTHRA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES59															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES59			Q1					RES60			Q0				
R-0h			R/W-10h					R-0h			R/W-10h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-57. EDMA_TPCC_QWMTHRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RES59	R	0h	RESERVE FIELD
12-8	Q1	R/W	10h	Queue Threshold for Q1 value
7-5	RES60	R	0h	RESERVE FIELD
4-0	Q0	R/W	10h	Queue Threshold for Q0 value

12.6.1.1.40 EDMA_TPCC_CCSTAT Register (Offset = 640h) [reset = 0h]

EDMA_TPCC_CCSTAT is shown in Figure 12-65 and described in Table 12-58.

Return to Summary Table.

CC Status Register

Figure 12-65. EDMA_TPCC_CCSTAT Register

31	30	29	28	27	26	25	24
RES61							
R-0h							
23	22	21	20	19	18	17	16
QUEACTV7	QUEACTV6	QUEACTV5	QUEACTV4	QUEACTV3	QUEACTV2	QUEACTV1	QUEACTV0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RES62		COMPACTV					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RES63			ACTV	RES64	TRACTV	QEVTACTV	EVTACTV
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-58. EDMA_TPCC_CCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RES61	R	0h	RESERVE FIELD
23	QUEACTV7	R	0h	Queue 7 Active QUEACTV7 = 0 : No Evts are queued in Q7. QUEACTV7 = 1 : At least one TR is queued in Q7.
22	QUEACTV6	R	0h	Queue 6 Active QUEACTV6 = 0 : No Evts are queued in Q6. QUEACTV6 = 1 : At least one TR is queued in Q6.
21	QUEACTV5	R	0h	Queue 5 Active QUEACTV5 = 0 : No Evts are queued in Q5. QUEACTV5 = 1 : At least one TR is queued in Q5.
20	QUEACTV4	R	0h	Queue 4 Active QUEACTV4 = 0 : No Evts are queued in Q4. QUEACTV4 = 1 : At least one TR is queued in Q4.
19	QUEACTV3	R	0h	Queue 3 Active QUEACTV3 = 0 : No Evts are queued in Q3. QUEACTV3 = 1 : At least one TR is queued in Q3.
18	QUEACTV2	R	0h	Queue 2 Active QUEACTV2 = 0 : No Evts are queued in Q2. QUEACTV2 = 1 : At least one TR is queued in Q2.
17	QUEACTV1	R	0h	Queue 1 Active QUEACTV1 = 0 : No Evts are queued in Q1. QUEACTV1 = 1 : At least one TR is queued in Q1.
16	QUEACTV0	R	0h	Queue 0 Active QUEACTV0 = 0 : No Evts are queued in Q0. QUEACTV0 = 1 : At least one TR is queued in Q0.
15-14	RES62	R	0h	RESERVE FIELD
13-8	COMPACTV	R	0h	Completion Request Active: Counter that tracks the total number of completion requests submitted to the TC. The counter increments when a TR is submitted with TCINTEN or TCCHEN set to '1'. The counter decrements for every valid completion code received from any of the external TCs. The CC will not service new TRs if COMPACTV count is already at the limit. COMPACTV = 0 : No completion requests outstanding. COMPACTV = 1: Total of '1' completion request outstanding. ... COMPACTV = 63 : Total of 63 completion requests are outstanding. No additional TRs will be submitted until count is less than 63.
7-5	RES63	R	0h	RESERVE FIELD

Table 12-58. EDMA_TPCC_CCSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACTV	R	0h	Channel Controller Active: Channel Controller Active is a logical-OR of each of the ACTV signals. The ACTV bit must remain high through the life of a TR. ACTV = 0 : Channel is idle. ACTV = 1 : Channel is busy.
3	RES64	R	0h	RESERVE FIELD
2	TRACTV	R	0h	Transfer Request Active: TRACTV = 0 : Transfer Request processing/submission logic is inactive. TRACTV = 1 : Transfer Request processing/submission logic is active.
1	QEVTACTV	R	0h	QDMA Event Active: QEVTACTV = 0 : No enabled QDMA Events are active within the CC. QEVTACTV = 1 : At least one enabled DMA Event (ER & EER ESR CER) is active within the CC.
0	EVTACTV	R	0h	DMA Event Active: EVTACTV = 0 : No enabled DMA Events are active within the CC. EVTACTV = 1 : At least one enabled DMA Event (ER & EER ESR CER) is active within the CC.

12.6.1.1.41 EDMA_TPCC_AETCTL Register (Offset = 700h) [reset = 0h]

EDMA_TPCC_AETCTL is shown in Figure 12-66 and described in Table 12-59.

Return to Summary Table.

Advanced Event Trigger Control

Figure 12-66. EDMA_TPCC_AETCTL Register

31	30	29	28	27	26	25	24
EN	RES65						
R/W-0h	R-0h						
23	22	21	20	19	18	17	16
RES65							
R-0h							
15	14	13	12	11	10	9	8
RES65		ENDINT					
R-0h		R/W-0h					
7	6	5	4	3	2	1	0
RES66	TYPE	STRTEVT					
R-0h	R/W-0h	R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-59. EDMA_TPCC_AETCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	EN	R/W	0h	AET Enable: EN = 0 : AET event generation is disabled. EN = 1 : AET event generation is enabled.
30-14	RES65	R	0h	RESERVE FIELD
13-8	ENDINT	R/W	0h	AET End Interrupt: Dictates the completion interrupt number that will force the tpcc_aet signal to be deasserted (low)
7	RES66	R	0h	RESERVE FIELD
6	TYPE	R/W	0h	AET Event Type: TYPE = 0 : Event specified by STARTEVT applies to DMA Events (set by ER ESR or CER) TYPE = 1 : Event specified by STARTEVT applies to QDMA Events
5-0	STRTEVT	R/W	0h	AET Start Event: Dictates the Event Number that will force the tpcc_aet signal to be asserted (high)

12.6.1.1.42 EDMA_TPCC_AETSTAT Register (Offset = 704h) [reset = 0h]

EDMA_TPCC_AETSTAT is shown in Figure 12-67 and described in Table 12-60.

Return to Summary Table.

Advanced Event Trigger Stat

Figure 12-67. EDMA_TPCC_AETSTAT Register

31	30	29	28	27	26	25	24
RES67							
R-0h							
23	22	21	20	19	18	17	16
RES67							
R-0h							
15	14	13	12	11	10	9	8
RES67							
R-0h							
7	6	5	4	3	2	1	0
RES67							STAT
R-0h							R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-60. EDMA_TPCC_AETSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RES67	R	0h	RESERVE FIELD
0	STAT	R	0h	AET Status: AETSTAT = 0 : tpcc_aet is currently low. AETSTAT = 1 : tpcc_aet is currently high.

12.6.1.1.43 EDMA_TPCC_AETCMD Register (Offset = 708h) [reset = 0h]

EDMA_TPCC_AETCMD is shown in Figure 12-68 and described in Table 12-61.

Return to Summary Table.

AET Command

Figure 12-68. EDMA_TPCC_AETCMD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES68															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES68															CLR
R-0h															W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-61. EDMA_TPCC_AETCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RES68	R	0h	RESERVE FIELD
0	CLR	W	0h	AET Clear command: CPU write of '1' to the CLR bit causes the tpcc_aet output signal and AETSTAT.STAT register to be cleared. CPU write of '0' has no effect..

12.6.1.1.44 EDMA_TPCC_ER Register (Offset = 1000h) [reset = 0h]

EDMA_TPCC_ER is shown in Figure 12-69 and described in Table 12-62.

Return to Summary Table.

Event Register: If ER.En bit is set and the EER.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ER.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EER.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EER register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECR pseudo-register.

Figure 12-69. EDMA_TPCC_ER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-62. EDMA_TPCC_ER Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8

Table 12-62. EDMA_TPCC_ER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.45 EDMA_TPCC_ERH Register (Offset = 1004h) [reset = 0h]

EDMA_TPCC_ERH is shown in Figure 12-70 and described in Table 12-63.

Return to Summary Table.

Event Register (High Part): If ERH.En bit is set and the EERH.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ERH.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EERH.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EERH register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECRH pseudo-register.

Figure 12-70. EDMA_TPCC_ERH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-63. EDMA_TPCC_ERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40

Table 12-63. EDMA_TPCC_ERH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.46 EDMA_TPCC_ECR Register (Offset = 1008h) [reset = 0h]

EDMA_TPCC_ECR is shown in Figure 12-71 and described in Table 12-64.

Return to Summary Table.

Event Clear Register: CPU write of '1' to the ECR.En bit causes the ER.En bit to be cleared. CPU write of '0' has no effect.

Figure 12-71. EDMA_TPCC_ECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-64. EDMA_TPCC_ECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-64. EDMA_TPCC_ECR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.47 EDMA_TPCC_ECRH Register (Offset = 100Ch) [reset = 0h]

EDMA_TPCC_ECRH is shown in Figure 12-72 and described in Table 12-65.

Return to Summary Table.

Event Clear Register (High Part): CPU write of '1' to the ECRH.En bit causes the ERH.En bit to be cleared. CPU write of '0' has no effect.

Figure 12-72. EDMA_TPCC_ECRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-65. EDMA_TPCC_ECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-65. EDMA_TPCC_ECRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.48 EDMA_TPCC_ESR Register (Offset = 1010h) [reset = 0h]

EDMA_TPCC_ESR is shown in Figure 12-73 and described in Table 12-66.

Return to Summary Table.

Event Set Register: CPU write of '1' to the ESR.En bit causes the ER.En bit to be set. CPU write of '0' has no effect.

Figure 12-73. EDMA_TPCC_ESR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-66. EDMA_TPCC_ESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-66. EDMA_TPCC_ESR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.49 EDMA_TPCC_ESRH Register (Offset = 1014h) [reset = 0h]

EDMA_TPCC_ESRH is shown in Figure 12-74 and described in Table 12-67.

Return to Summary Table.

Event Set Register (High Part) CPU write of '1' to the ESRH.En bit causes the ERH.En bit to be set. CPU write of '0' has no effect.

Figure 12-74. EDMA_TPCC_ESRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-67. EDMA_TPCC_ESRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-67. EDMA_TPCC_ESRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.50 EDMA_TPCC_CER Register (Offset = 1018h) [reset = 0h]

EDMA_TPCC_CER is shown in Figure 12-75 and described in Table 12-68.

Return to Summary Table.

Chained Event Register: If CER.En bit is set (regardless of state of EER.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CER.En bit is cleared when the corresponding event is prioritized and serviced. If the CER.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CER.En cannot be set or cleared via software.

Figure 12-75. EDMA_TPCC_CER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-68. EDMA_TPCC_CER Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8

Table 12-68. EDMA_TPCC_CER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.51 EDMA_TPCC_CERH Register (Offset = 101Ch) [reset = 0h]

EDMA_TPCC_CERH is shown in Figure 12-76 and described in Table 12-69.

Return to Summary Table.

Chained Event Register (High Part): If CERH.En bit is set (regardless of state of EERH.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the CERH.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CERH.En cannot be set or cleared via software.

Figure 12-76. EDMA_TPCC_CERH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-69. EDMA_TPCC_CERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40

Table 12-69. EDMA_TPCC_CERH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.52 EDMA_TPCC_EER Register (Offset = 1020h) [reset = 0h]

EDMA_TPCC_EER is shown in Figure 12-77 and described in Table 12-70.

Return to Summary Table.

Event Enable Register: Enables DMA transfers for ER.En pending events. ER.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (CER) or Event Set Register (ESR). Note that if a bit is set in ER.En while EER.En is disabled no action is taken. If EER.En is enabled at a later point (and ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EER.En is not directly writeable. Events can be enabled via writes to EESR and can be disabled via writes to EECR register. EER.En = 0: ER.En is not enabled to trigger DMA transfers. EER.En = 1: ER.En is enabled to trigger DMA transfers.

Figure 12-77. EDMA_TPCC_EER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-70. EDMA_TPCC_EER Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8

Table 12-70. EDMA_TPCC_EER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.53 EDMA_TPCC_EERH Register (Offset = 1024h) [reset = 0h]

EDMA_TPCC_EERH is shown in Figure 12-78 and described in Table 12-71.

Return to Summary Table.

Event Enable Register (High Part): Enables DMA transfers for ERH.En pending events. ERH.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (CERH) or Event Set Register (ESRH). Note that if a bit is set in ERH.En while EERH.En is disabled no action is taken. If EERH.En is enabled at a later point (and ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EERH.En is not directly writeable. Events can be enabled via writes to EESRH and can be disabled via writes to EECRH register. EERH.En = 0: ER.En is not enabled to trigger DMA transfers. EERH.En = 1: ER.En is enabled to trigger DMA transfers.

Figure 12-78. EDMA_TPCC_EERH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-71. EDMA_TPCC_EERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40

Table 12-71. EDMA_TPCC_EERH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.54 EDMA_TPCC_EECR Register (Offset = 1028h) [reset = 0h]

EDMA_TPCC_EECR is shown in Figure 12-79 and described in Table 12-72.

Return to Summary Table.

Event Enable Clear Register: CPU write of '1' to the EECR.En bit causes the EER.En bit to be cleared. CPU write of '0' has no effect..

Figure 12-79. EDMA_TPCC_EECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-72. EDMA_TPCC_EECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-72. EDMA_TPCC_EECR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.55 EDMA_TPCC_EECRH Register (Offset = 102Ch) [reset = 0h]

EDMA_TPCC_EECRH is shown in Figure 12-80 and described in Table 12-73.

Return to Summary Table.

Event Enable Clear Register (High Part): CPU write of '1' to the EECRH.En bit causes the EERH.En bit to be cleared. CPU write of '0' has no effect..

Figure 12-80. EDMA_TPCC_EECRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-73. EDMA_TPCC_EECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-73. EDMA_TPCC_EECRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.56 EDMA_TPCC_EESR Register (Offset = 1030h) [reset = 0h]

EDMA_TPCC_EESR is shown in Figure 12-81 and described in Table 12-74.

Return to Summary Table.

Event Enable Set Register: CPU write of '1' to the EESR.En bit causes the EER.En bit to be set. CPU write of '0' has no effect..

Figure 12-81. EDMA_TPCC_EESR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-74. EDMA_TPCC_EESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-74. EDMA_TPCC_EESR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.57 EDMA_TPCC_EESRH Register (Offset = 1034h) [reset = 0h]

EDMA_TPCC_EESRH is shown in Figure 12-82 and described in Table 12-75.

Return to Summary Table.

Event Enable Set Register (High Part): CPU write of '1' to the EESRH.En bit causes the EERH.En bit to be set. CPU write of '0' has no effect..

Figure 12-82. EDMA_TPCC_EESRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-75. EDMA_TPCC_EESRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-75. EDMA_TPCC_EESRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.58 EDMA_TPCC_SER Register (Offset = 1038h) [reset = 0h]

EDMA_TPCC_SER is shown in Figure 12-83 and described in Table 12-76.

Return to Summary Table.

Secondary Event Register: The secondary event register is used along with the Event Register (ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

Figure 12-83. EDMA_TPCC_SER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-76. EDMA_TPCC_SER Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5

Table 12-76. EDMA_TPCC_SER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.59 EDMA_TPCC_SERH Register (Offset = 103Ch) [reset = 0h]

EDMA_TPCC_SERH is shown in Figure 12-84 and described in Table 12-77.

Return to Summary Table.

Secondary Event Register (High Part): The secondary event register is used along with the Event Register (ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

Figure 12-84. EDMA_TPCC_SERH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-77. EDMA_TPCC_SERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37

Table 12-77. EDMA_TPCC_SERH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.60 EDMA_TPCC_SECR Register (Offset = 1040h) [reset = 0h]

EDMA_TPCC_SECR is shown in Figure 12-85 and described in Table 12-78.

Return to Summary Table.

Secondary Event Clear Register: The secondary event clear register is used to clear the status of the SER registers. CPU write of '1' to the SECR.En bit clears the SER register. CPU write of '0' has no effect.

Figure 12-85. EDMA_TPCC_SECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-78. EDMA_TPCC_SECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-78. EDMA_TPCC_SECR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.61 EDMA_TPCC_SECRH Register (Offset = 1044h) [reset = 0h]

EDMA_TPCC_SECRH is shown in Figure 12-86 and described in Table 12-79.

Return to Summary Table.

Secondary Event Clear Register (High Part): The secondary event clear register is used to clear the status of the SERH registers. CPU write of '1' to the SECRH.En bit clears the SERH register. CPU write of '0' has no effect.

Figure 12-86. EDMA_TPCC_SECRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-79. EDMA_TPCC_SECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37

Table 12-79. EDMA_TPCC_SECRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.62 EDMA_TPCC_IER Register (Offset = 1050h) [reset = 0h]

EDMA_TPCC_IER is shown in Figure 12-87 and described in Table 12-80.

Return to Summary Table.

Int Enable Register: IER.In is not directly writeable. Interrupts can be enabled via writes to IESR and can be disabled via writes to IECR register. IER.In = 0: IPR.In is NOT enabled for interrupts. IER.In = 1: IPR.In IS enabled for interrupts.

Figure 12-87. EDMA_TPCC_IER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-80. EDMA_TPCC_IER Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6
5	I5	R	0h	Interrupt associated with TCC #5

Table 12-80. EDMA_TPCC_IER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	I4	R	0h	Interrupt associated with TCC #4
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0

12.6.1.1.63 EDMA_TPCC_IERH Register (Offset = 1054h) [reset = 0h]

EDMA_TPCC_IERH is shown in Figure 12-88 and described in Table 12-81.

Return to Summary Table.

Int Enable Register (High Part): IERH.In is not directly writeable. Interrupts can be enabled via writes to IESRH and can be disabled via writes to IECRH register. IERH.In = 0: IPRH.In is NOT enabled for interrupts. IERH.In = 1: IPRH.In IS enabled for interrupts.

Figure 12-88. EDMA_TPCC_IERH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-81. EDMA_TPCC_IERH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38
5	I37	R	0h	Interrupt associated with TCC #37

Table 12-81. EDMA_TPCC_IERH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	I36	R	0h	Interrupt associated with TCC #36
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

12.6.1.1.64 EDMA_TPCC_IECR Register (Offset = 1058h) [reset = 0h]

EDMA_TPCC_IECR is shown in Figure 12-89 and described in Table 12-82.

Return to Summary Table.

Int Enable Clear Register: CPU write of '1' to the IECR.In bit causes the IER.In bit to be cleared. CPU write of '0' has no effect..

Figure 12-89. EDMA_TPCC_IECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-82. EDMA_TPCC_IECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4

Table 12-82. EDMA_TPCC_IECR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

12.6.1.1.65 EDMA_TPCC_IECRH Register (Offset = 105Ch) [reset = 0h]

EDMA_TPCC_IECRH is shown in Figure 12-90 and described in Table 12-83.

Return to Summary Table.

Int Enable Clear Register (High Part): CPU write of '1' to the IECRH.In bit causes the IERH.In bit to be cleared. CPU write of '0' has no effect..

Figure 12-90. EDMA_TPCC_IECRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-83. EDMA_TPCC_IECRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36

Table 12-83. EDMA_TPCC_IECRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

12.6.1.1.66 EDMA_TPCC_IESR Register (Offset = 1060h) [reset = 0h]

EDMA_TPCC_IESR is shown in Figure 12-91 and described in Table 12-84.

Return to Summary Table.

Int Enable Set Register: CPU write of '1' to the IESR.In bit causes the IESR.In bit to be set. CPU write of '0' has no effect..

Figure 12-91. EDMA_TPCC_IESR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-84. EDMA_TPCC_IESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4

Table 12-84. EDMA_TPCC_IESR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

12.6.1.1.67 EDMA_TPCC_IESRH Register (Offset = 1064h) [reset = 0h]

EDMA_TPCC_IESRH is shown in Figure 12-92 and described in Table 12-85.

Return to Summary Table.

Int Enable Set Register (High Part): CPU write of '1' to the IESRH.In bit causes the IESRH.In bit to be set. CPU write of '0' has no effect..

Figure 12-92. EDMA_TPCC_IESRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-85. EDMA_TPCC_IESRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36

Table 12-85. EDMA_TPCC_IESRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

12.6.1.1.68 EDMA_TPCC_IPR Register (Offset = 1068h) [reset = 0h]

EDMA_TPCC_IPR is shown in Figure 12-93 and described in Table 12-86.

Return to Summary Table.

Interrupt Pending Register: IPR.In bit is set when a interrupt completion code with TCC of N is detected. IPR.In bit is cleared via software by writing a '1' to ICR.In bit.

Figure 12-93. EDMA_TPCC_IPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-86. EDMA_TPCC_IPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6
5	I5	R	0h	Interrupt associated with TCC #5
4	I4	R	0h	Interrupt associated with TCC #4

Table 12-86. EDMA_TPCC_IPR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0

12.6.1.1.69 EDMA_TPCC_IPRH Register (Offset = 106Ch) [reset = 0h]

EDMA_TPCC_IPRH is shown in Figure 12-94 and described in Table 12-87.

Return to Summary Table.

Interrupt Pending Register (High Part): IPRH.In bit is set when a interrupt completion code with TCC of N is detected. IPRH.In bit is cleared via software by writing a '1' to ICRH.In bit.

Figure 12-94. EDMA_TPCC_IPRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-87. EDMA_TPCC_IPRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38
5	I37	R	0h	Interrupt associated with TCC #37
4	I36	R	0h	Interrupt associated with TCC #36

Table 12-87. EDMA_TPCC_IPRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

12.6.1.1.70 EDMA_TPCC_ICR Register (Offset = 1070h) [reset = 0h]

EDMA_TPCC_ICR is shown in Figure 12-95 and described in Table 12-88.

Return to Summary Table.

Interrupt Clear Register: CPU write of '1' to the ICR.In bit causes the IPR.In bit to be cleared. CPU write of '0' has no effect. All IPR.In bits must be cleared before additional interrupts will be asserted by CC.

Figure 12-95. EDMA_TPCC_ICR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-88. EDMA_TPCC_ICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4

Table 12-88. EDMA_TPCC_ICR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

12.6.1.1.71 EDMA_TPCC_ICRH Register (Offset = 1074h) [reset = 0h]

EDMA_TPCC_ICRH is shown in Figure 12-96 and described in Table 12-89.

Return to Summary Table.

Interrupt Clear Register (High Part): CPU write of '1' to the ICRH.In bit causes the IPRH.In bit to be cleared. CPU write of '0' has no effect. All IPRH.In bits must be cleared before additional interrupts will be asserted by CC.

Figure 12-96. EDMA_TPCC_ICRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-89. EDMA_TPCC_ICRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38
5	I37	W	0h	Interrupt associated with TCC #37

Table 12-89. EDMA_TPCC_ICRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

12.6.1.1.72 EDMA_TPCC_IEVAL Register (Offset = 1078h) [reset = 0h]

EDMA_TPCC_IEVAL is shown in Figure 12-97 and described in Table 12-90.

Return to Summary Table.

Interrupt Eval Register

Figure 12-97. EDMA_TPCC_IEVAL Register

31	30	29	28	27	26	25	24
RES69							
R-0h							
23	22	21	20	19	18	17	16
RES69							
R-0h							
15	14	13	12	11	10	9	8
RES69							
R-0h							
7	6	5	4	3	2	1	0
RES69						SET	EVAL
R-0h						W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-90. EDMA_TPCC_IEVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES69	R	0h	RESERVE FIELD
1	SET	W	0h	Interrupt Set: CPU write of '1' to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (IPRn). CPU write of '0' has no effect.
0	EVAL	W	0h	Interrupt Evaluate: CPU write of '1' to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (IPRn). CPU write of '0' has no effect..

12.6.1.1.73 EDMA_TPCC_QER Register (Offset = 1080h) [reset = 0h]

EDMA_TPCC_QER is shown in Figure 12-98 and described in Table 12-91.

Return to Summary Table.

QDMA Event Register: If QER.En bit is set then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. QER.En bit is cleared when the corresponding event is prioritized and serviced. QER.En is also cleared when user writes a '1' to the QSECR.En bit. If the QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and QEER register is set then the corresponding bit in the QDMA Event Missed Register is set.

Figure 12-98. EDMA_TPCC_QER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES70															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES70								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-91. EDMA_TPCC_QER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES70	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.74 EDMA_TPCC_QEER Register (Offset = 1084h) [reset = 0h]

EDMA_TPCC_QEER is shown in Figure 12-99 and described in Table 12-92.

Return to Summary Table.

QDMA Event Enable Register: Enabled/disabled QDMA address comparator for QDMA Channel N. QEER.En is not directly writeable. QDMA channels can be enabled via writes to QEESR and can be disabled via writes to QEECR register. QEER.En = 1 The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in QER.En. QEER.En = 0 The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in QER.En.

Figure 12-99. EDMA_TPCC_QEER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES71															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES71								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-92. EDMA_TPCC_QEER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES71	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.75 EDMA_TPCC_QEECR Register (Offset = 1088h) [reset = 0h]

EDMA_TPCC_QEECR is shown in Figure 12-100 and described in Table 12-93.

Return to Summary Table.

QDMA Event Enable Clear Register: CPU write of '1' to the QEECR.En bit causes the QEER.En bit to be cleared. CPU write of '0' has no effect..

Figure 12-100. EDMA_TPCC_QEECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES72															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES72								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-93. EDMA_TPCC_QEECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES72	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.76 EDMA_TPCC_QEESR Register (Offset = 108Ch) [reset = 0h]

EDMA_TPCC_QEESR is shown in Figure 12-101 and described in Table 12-94.

Return to Summary Table.

QDMA Event Enable Set Register: CPU write of '1' to the QEESR.En bit causes the QEESR.En bit to be set. CPU write of '0' has no effect..

Figure 12-101. EDMA_TPCC_QEESR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES73															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES73								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-94. EDMA_TPCC_QEESR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES73	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.77 EDMA_TPCC_QSER Register (Offset = 1090h) [reset = 0h]

EDMA_TPCC_QSER is shown in Figure 12-102 and described in Table 12-95.

Return to Summary Table.

QDMA Secondary Event Register: The QDMA secondary event register is used along with the QDMA Event Register (QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

Figure 12-102. EDMA_TPCC_QSER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES74															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES74								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-95. EDMA_TPCC_QSER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES74	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.78 EDMA_TPCC_QSECR Register (Offset = 1094h) [reset = 0h]

EDMA_TPCC_QSECR is shown in Figure 12-103 and described in Table 12-96.

Return to Summary Table.

QDMA Secondary Event Clear Register: The secondary event clear register is used to clear the status of the QSER and QER register (note that this is slightly different than the SER operation which does not clear the ER.En register). CPU write of '1' to the QSECR.En bit clears the QSER.En and QER.En register fields. CPU write of '0' has no effect..

Figure 12-103. EDMA_TPCC_QSECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES75															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES75								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-96. EDMA_TPCC_QSECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES75	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.79 EDMA_TPCC_SHADOW_N Register (Offset = 2000h)

EDMA_TPCC_SHADOW_N is shown in Figure 12-104 and described in Table 12-97.

Return to Summary Table.

Bundle description is not available

Figure 12-104. EDMA_TPCC_SHADOW_N Register

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-97. EDMA_TPCC_SHADOW_N Register Field Descriptions

Bit	Field	Type	Reset	Description

12.6.1.1.80 EDMA_TPCC_ER_RN Register (Offset = 2000h) [reset = 0h]

EDMA_TPCC_ER_RN is shown in Figure 12-105 and described in Table 12-98.

Return to Summary Table.

Event Register: If ER.En bit is set and the EER.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ER.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EER.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EER register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECR pseudo-register.

Figure 12-105. EDMA_TPCC_ER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-98. EDMA_TPCC_ER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8

Table 12-98. EDMA_TPCC_ER_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.81 EDMA_TPCC_ERH_RN Register (Offset = 2004h) [reset = 0h]

EDMA_TPCC_ERH_RN is shown in Figure 12-106 and described in Table 12-99.

Return to Summary Table.

Event Register (High Part): If ERH.En bit is set and the EERH.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ERH.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EERH.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EERH register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECRH pseudo-register.

Figure 12-106. EDMA_TPCC_ERH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-99. EDMA_TPCC_ERH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40

Table 12-99. EDMA_TPCC_ERH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.82 EDMA_TPCC_ECR_RN Register (Offset = 2008h) [reset = 0h]

EDMA_TPCC_ECR_RN is shown in Figure 12-107 and described in Table 12-100.

Return to Summary Table.

Event Clear Register: CPU write of '1' to the ECR.En bit causes the ER.En bit to be cleared. CPU write of '0' has no effect.

Figure 12-107. EDMA_TPCC_ECR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-100. EDMA_TPCC_ECR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-100. EDMA_TPCC_ECR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.83 EDMA_TPCC_ECRH_RN Register (Offset = 200Ch) [reset = 0h]

EDMA_TPCC_ECRH_RN is shown in Figure 12-108 and described in Table 12-101.

Return to Summary Table.

Event Clear Register (High Part): CPU write of '1' to the ECRH.En bit causes the ERH.En bit to be cleared. CPU write of '0' has no effect.

Figure 12-108. EDMA_TPCC_ECRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-101. EDMA_TPCC_ECRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-101. EDMA_TPCC_ECRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.84 EDMA_TPCC_ESR_RN Register (Offset = 2010h) [reset = 0h]

EDMA_TPCC_ESR_RN is shown in Figure 12-109 and described in Table 12-102.

Return to Summary Table.

Event Set Register: CPU write of '1' to the ESR.En bit causes the ER.En bit to be set. CPU write of '0' has no effect.

Figure 12-109. EDMA_TPCC_ESR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-102. EDMA_TPCC_ESR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-102. EDMA_TPCC_ESR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.85 EDMA_TPCC_ESRH_RN Register (Offset = 2014h) [reset = 0h]

EDMA_TPCC_ESRH_RN is shown in Figure 12-110 and described in Table 12-103.

Return to Summary Table.

Event Set Register (High Part) CPU write of '1' to the ESRH.En bit causes the ERH.En bit to be set. CPU write of '0' has no effect.

Figure 12-110. EDMA_TPCC_ESRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-103. EDMA_TPCC_ESRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-103. EDMA_TPCC_ESRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.86 EDMA_TPCC_CER_RN Register (Offset = 2018h) [reset = 0h]

EDMA_TPCC_CER_RN is shown in Figure 12-111 and described in Table 12-104.

Return to Summary Table.

Chained Event Register: If CER.En bit is set (regardless of state of EER.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CER.En bit is cleared when the corresponding event is prioritized and serviced. If the CER.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CER.En cannot be set or cleared via software.

Figure 12-111. EDMA_TPCC_CER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-104. EDMA_TPCC_CER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8

Table 12-104. EDMA_TPCC_CER_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.87 EDMA_TPCC_CERH_RN Register (Offset = 201Ch) [reset = 0h]

EDMA_TPCC_CERH_RN is shown in Figure 12-112 and described in Table 12-105.

Return to Summary Table.

Chained Event Register (High Part): If CERH.En bit is set (regardless of state of EERH.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the CERH.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CERH.En cannot be set or cleared via software.

Figure 12-112. EDMA_TPCC_CERH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-105. EDMA_TPCC_CERH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40

Table 12-105. EDMA_TPCC_CERH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.88 EDMA_TPCC_EER_RN Register (Offset = 2020h) [reset = 0h]

EDMA_TPCC_EER_RN is shown in Figure 12-113 and described in Table 12-106.

Return to Summary Table.

Event Enable Register: Enables DMA transfers for ER.En pending events. ER.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (CER) or Event Set Register (ESR). Note that if a bit is set in ER.En while EER.En is disabled no action is taken. If EER.En is enabled at a later point (and ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EER.En is not directly writeable. Events can be enabled via writes to EESR and can be disabled via writes to EECR register. EER.En = 0: ER.En is not enabled to trigger DMA transfers. EER.En = 1: ER.En is enabled to trigger DMA transfers.

Figure 12-113. EDMA_TPCC_EER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-106. EDMA_TPCC_EER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8

Table 12-106. EDMA_TPCC_EER_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.89 EDMA_TPCC_EERH_RN Register (Offset = 2024h) [reset = 0h]

EDMA_TPCC_EERH_RN is shown in Figure 12-114 and described in Table 12-107.

Return to Summary Table.

Event Enable Register (High Part): Enables DMA transfers for ERH.En pending events. ERH.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (CERH) or Event Set Register (ESRH). Note that if a bit is set in ERH.En while EERH.En is disabled no action is taken. If EERH.En is enabled at a later point (and ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EERH.En is not directly writeable. Events can be enabled via writes to EESRH and can be disabled via writes to EECRH register. EERH.En = 0: ER.En is not enabled to trigger DMA transfers. EERH.En = 1: ER.En is enabled to trigger DMA transfers.

Figure 12-114. EDMA_TPCC_EERH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-107. EDMA_TPCC_EERH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40

Table 12-107. EDMA_TPCC_EERH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.90 EDMA_TPCC_EECR_RN Register (Offset = 2028h) [reset = 0h]

EDMA_TPCC_EECR_RN is shown in Figure 12-115 and described in Table 12-108.

Return to Summary Table.

Event Enable Clear Register: CPU write of '1' to the EECR.En bit causes the EER.En bit to be cleared. CPU write of '0' has no effect..

Figure 12-115. EDMA_TPCC_EECR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-108. EDMA_TPCC_EECR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-108. EDMA_TPCC_EECR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.91 EDMA_TPCC_EECRH_RN Register (Offset = 202Ch) [reset = 0h]

EDMA_TPCC_EECRH_RN is shown in Figure 12-116 and described in Table 12-109.

Return to Summary Table.

Event Enable Clear Register (High Part): CPU write of '1' to the EECRH.En bit causes the EERH.En bit to be cleared. CPU write of '0' has no effect..

Figure 12-116. EDMA_TPCC_EECRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-109. EDMA_TPCC_EECRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-109. EDMA_TPCC_EECRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.92 EDMA_TPCC_EESR_RN Register (Offset = 2030h) [reset = 0h]

EDMA_TPCC_EESR_RN is shown in Figure 12-117 and described in Table 12-110.

Return to Summary Table.

Event Enable Set Register: CPU write of '1' to the EESR.En bit causes the EER.En bit to be set. CPU write of '0' has no effect..

Figure 12-117. EDMA_TPCC_EESR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-110. EDMA_TPCC_EESR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-110. EDMA_TPCC_EESR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.93 EDMA_TPCC_EESRH_RN Register (Offset = 2034h) [reset = 0h]

EDMA_TPCC_EESRH_RN is shown in Figure 12-118 and described in Table 12-111.

Return to Summary Table.

Event Enable Set Register (High Part): CPU write of '1' to the EESRH.En bit causes the EERH.En bit to be set. CPU write of '0' has no effect..

Figure 12-118. EDMA_TPCC_EESRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-111. EDMA_TPCC_EESRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36

Table 12-111. EDMA_TPCC_EESRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.94 EDMA_TPCC_SER_RN Register (Offset = 2038h) [reset = 0h]

EDMA_TPCC_SER_RN is shown in Figure 12-119 and described in Table 12-112.

Return to Summary Table.

Secondary Event Register: The secondary event register is used along with the Event Register (ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

Figure 12-119. EDMA_TPCC_SER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-112. EDMA_TPCC_SER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5

Table 12-112. EDMA_TPCC_SER_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.95 EDMA_TPCC_SERH_RN Register (Offset = 203Ch) [reset = 0h]

EDMA_TPCC_SERH_RN is shown in Figure 12-120 and described in Table 12-113.

Return to Summary Table.

Secondary Event Register (High Part): The secondary event register is used along with the Event Register (ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

Figure 12-120. EDMA_TPCC_SERH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-113. EDMA_TPCC_SERH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37

Table 12-113. EDMA_TPCC_SERH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

12.6.1.1.96 EDMA_TPCC_SECR_RN Register (Offset = 2040h) [reset = 0h]

EDMA_TPCC_SECR_RN is shown in Figure 12-121 and described in Table 12-114.

Return to Summary Table.

Secondary Event Clear Register: The secondary event clear register is used to clear the status of the SER registers. CPU write of '1' to the SECR.En bit clears the SER register. CPU write of '0' has no effect.

Figure 12-121. EDMA_TPCC_SECR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-114. EDMA_TPCC_SECR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4

Table 12-114. EDMA_TPCC_SECR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.97 EDMA_TPCC_SECRH_RN Register (Offset = 2044h) [reset = 0h]

EDMA_TPCC_SECRH_RN is shown in Figure 12-122 and described in Table 12-115.

Return to Summary Table.

Secondary Event Clear Register (High Part): The secondary event clear register is used to clear the status of the SERH registers. CPU write of '1' to the SECRH.En bit clears the SERH register. CPU write of '0' has no effect.

Figure 12-122. EDMA_TPCC_SECRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-115. EDMA_TPCC_SECRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38
5	E37	W	0h	Event #37

Table 12-115. EDMA_TPCC_SECRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

12.6.1.1.98 EDMA_TPCC_IER_RN Register (Offset = 2050h) [reset = 0h]

EDMA_TPCC_IER_RN is shown in Figure 12-123 and described in Table 12-116.

Return to Summary Table.

Int Enable Register: IER.In is not directly writeable. Interrupts can be enabled via writes to IESR and can be disabled via writes to IECR register. IER.In = 0: IPR.In is NOT enabled for interrupts. IER.In = 1: IPR.In IS enabled for interrupts.

Figure 12-123. EDMA_TPCC_IER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-116. EDMA_TPCC_IER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6
5	I5	R	0h	Interrupt associated with TCC #5

Table 12-116. EDMA_TPCC_IER_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	I4	R	0h	Interrupt associated with TCC #4
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0

12.6.1.1.99 EDMA_TPCC_IERH_RN Register (Offset = 2054h) [reset = 0h]

EDMA_TPCC_IERH_RN is shown in Figure 12-124 and described in Table 12-117.

Return to Summary Table.

Int Enable Register (High Part): IERH.In is not directly writeable. Interrupts can be enabled via writes to IESRH and can be disabled via writes to IECRH register. IERH.In = 0: IPRH.In is NOT enabled for interrupts. IERH.In = 1: IPRH.In IS enabled for interrupts.

Figure 12-124. EDMA_TPCC_IERH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-117. EDMA_TPCC_IERH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38
5	I37	R	0h	Interrupt associated with TCC #37

Table 12-117. EDMA_TPCC_IERH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	I36	R	0h	Interrupt associated with TCC #36
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

12.6.1.1.100 EDMA_TPCC_IECR_RN Register (Offset = 2058h) [reset = 0h]

EDMA_TPCC_IECR_RN is shown in Figure 12-125 and described in Table 12-118.

Return to Summary Table.

Int Enable Clear Register: CPU write of '1' to the IECR.In bit causes the IER.In bit to be cleared. CPU write of '0' has no effect..

Figure 12-125. EDMA_TPCC_IECR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-118. EDMA_TPCC_IECR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4

Table 12-118. EDMA_TPCC_IECR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

12.6.1.1.101 EDMA_TPCC_IECRH_RN Register (Offset = 205Ch) [reset = 0h]

EDMA_TPCC_IECRH_RN is shown in Figure 12-126 and described in Table 12-119.

Return to Summary Table.

Int Enable Clear Register (High Part): CPU write of '1' to the IECRH.In bit causes the IERH.In bit to be cleared. CPU write of '0' has no effect..

Figure 12-126. EDMA_TPCC_IECRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-119. EDMA_TPCC_IECRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36

Table 12-119. EDMA_TPCC_IECRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

12.6.1.1.102 EDMA_TPCC_IESR_RN Register (Offset = 2060h) [reset = 0h]

EDMA_TPCC_IESR_RN is shown in Figure 12-127 and described in Table 12-120.

Return to Summary Table.

Int Enable Set Register: CPU write of '1' to the IESR.In bit causes the IESR.In bit to be set. CPU write of '0' has no effect..

Figure 12-127. EDMA_TPCC_IESR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-120. EDMA_TPCC_IESR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4

Table 12-120. EDMA_TPCC_IESR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

12.6.1.1.103 EDMA_TPCC_IESRH_RN Register (Offset = 2064h) [reset = 0h]

EDMA_TPCC_IESRH_RN is shown in Figure 12-128 and described in Table 12-121.

Return to Summary Table.

Int Enable Set Register (High Part): CPU write of '1' to the IESRH.In bit causes the IESRH.In bit to be set. CPU write of '0' has no effect..

Figure 12-128. EDMA_TPCC_IESRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-121. EDMA_TPCC_IESRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36

Table 12-121. EDMA_TPCC_IESRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

12.6.1.1.104 EDMA_TPCC_IPR_RN Register (Offset = 2068h) [reset = 0h]

EDMA_TPCC_IPR_RN is shown in Figure 12-129 and described in Table 12-122.

Return to Summary Table.

Interrupt Pending Register: IPR.In bit is set when a interrupt completion code with TCC of N is detected. IPR.In bit is cleared via software by writing a '1' to ICR.In bit.

Figure 12-129. EDMA_TPCC_IPR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-122. EDMA_TPCC_IPR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6
5	I5	R	0h	Interrupt associated with TCC #5
4	I4	R	0h	Interrupt associated with TCC #4

Table 12-122. EDMA_TPCC_IPR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0

12.6.1.1.105 EDMA_TPCC_IPRH_RN Register (Offset = 206Ch) [reset = 0h]

EDMA_TPCC_IPRH_RN is shown in Figure 12-130 and described in Table 12-123.

Return to Summary Table.

Interrupt Pending Register (High Part): IPRH.In bit is set when a interrupt completion code with TCC of N is detected. IPRH.In bit is cleared via software by writing a '1' to ICRH.In bit.

Figure 12-130. EDMA_TPCC_IPRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-123. EDMA_TPCC_IPRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38
5	I37	R	0h	Interrupt associated with TCC #37
4	I36	R	0h	Interrupt associated with TCC #36

Table 12-123. EDMA_TPCC_IPRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

12.6.1.1.106 EDMA_TPCC_ICR_RN Register (Offset = 2070h) [reset = 0h]

EDMA_TPCC_ICR_RN is shown in Figure 12-131 and described in Table 12-124.

Return to Summary Table.

Interrupt Clear Register: CPU write of '1' to the ICR.In bit causes the IPR.In bit to be cleared. CPU write of '0' has no effect. All IPR.In bits must be cleared before additional interrupts will be asserted by CC.

Figure 12-131. EDMA_TPCC_ICR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-124. EDMA_TPCC_ICR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4

Table 12-124. EDMA_TPCC_ICR_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

12.6.1.1.107 EDMA_TPCC_ICRH_RN Register (Offset = 2074h) [reset = 0h]

EDMA_TPCC_ICRH_RN is shown in Figure 12-132 and described in Table 12-125.

Return to Summary Table.

Interrupt Clear Register (High Part): CPU write of '1' to the ICRH.In bit causes the IPRH.In bit to be cleared. CPU write of '0' has no effect. All IPRH.In bits must be cleared before additional interrupts will be asserted by CC.

Figure 12-132. EDMA_TPCC_ICRH_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-125. EDMA_TPCC_ICRH_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38
5	I37	W	0h	Interrupt associated with TCC #37

Table 12-125. EDMA_TPCC_ICRH_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

12.6.1.1.108 EDMA_TPCC_IEVAL_RN Register (Offset = 2078h) [reset = 0h]

EDMA_TPCC_IEVAL_RN is shown in Figure 12-133 and described in Table 12-126.

Return to Summary Table.

Interrupt Eval Register

Figure 12-133. EDMA_TPCC_IEVAL_RN Register

31	30	29	28	27	26	25	24
RES76							
R-0h							
23	22	21	20	19	18	17	16
RES76							
R-0h							
15	14	13	12	11	10	9	8
RES76							
R-0h							
7	6	5	4	3	2	1	0
RES76						SET	EVAL
R-0h						W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-126. EDMA_TPCC_IEVAL_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES76	R	0h	RESERVE FIELD
1	SET	W	0h	Interrupt Set: CPU write of '1' to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (IPRn). CPU write of '0' has no effect.
0	EVAL	W	0h	Interrupt Evaluate: CPU write of '1' to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (IPRn). CPU write of '0' has no effect..

12.6.1.1.109 EDMA_TPCC_QER_RN Register (Offset = 2080h) [reset = 0h]

EDMA_TPCC_QER_RN is shown in Figure 12-134 and described in Table 12-127.

Return to Summary Table.

QDMA Event Register: If QER.En bit is set then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. QER.En bit is cleared when the corresponding event is prioritized and serviced. QER.En is also cleared when user writes a '1' to the QSECR.En bit. If the QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and QEER register is set then the corresponding bit in the QDMA Event Missed Register is set.

Figure 12-134. EDMA_TPCC_QER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES77															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES77								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-127. EDMA_TPCC_QER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES77	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.110 EDMA_TPCC_QEER_RN Register (Offset = 2084h) [reset = 0h]

EDMA_TPCC_QEER_RN is shown in Figure 12-135 and described in Table 12-128.

Return to Summary Table.

QDMA Event Enable Register: Enabled/disabled QDMA address comparator for QDMA Channel N. QEER.En is not directly writeable. QDMA channels can be enabled via writes to QEESR and can be disabled via writes to QEECR register. QEER.En = 1 The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in QER.En. QEER.En = 0 The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in QER.En.

Figure 12-135. EDMA_TPCC_QEER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES78															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES78								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-128. EDMA_TPCC_QEER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES78	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.111 EDMA_TPCC_QEECR_RN Register (Offset = 2088h) [reset = 0h]

EDMA_TPCC_QEECR_RN is shown in Figure 12-136 and described in Table 12-129.

Return to Summary Table.

QDMA Event Enable Clear Register: CPU write of '1' to the QEECR.En bit causes the QEER.En bit to be cleared. CPU write of '0' has no effect..

Figure 12-136. EDMA_TPCC_QEECR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES79															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES79								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-129. EDMA_TPCC_QEECR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES79	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.112 EDMA_TPCC_QEESR_RN Register (Offset = 208Ch) [reset = 0h]

EDMA_TPCC_QEESR_RN is shown in Figure 12-137 and described in Table 12-130.

Return to Summary Table.

QDMA Event Enable Set Register: CPU write of '1' to the QEESR.En bit causes the QEESR.En bit to be set. CPU write of '0' has no effect..

Figure 12-137. EDMA_TPCC_QEESR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES80															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES80								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-130. EDMA_TPCC_QEESR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES80	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.113 EDMA_TPCC_QSER_RN Register (Offset = 2090h) [reset = 0h]

EDMA_TPCC_QSER_RN is shown in Figure 12-138 and described in Table 12-131.

Return to Summary Table.

QDMA Secondary Event Register: The QDMA secondary event register is used along with the QDMA Event Register (QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

Figure 12-138. EDMA_TPCC_QSER_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES81															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES81								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-131. EDMA_TPCC_QSER_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES81	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

12.6.1.1.114 EDMA_TPCC_QSECR_RN Register (Offset = 2094h) [reset = 0h]

EDMA_TPCC_QSECR_RN is shown in Figure 12-139 and described in Table 12-132.

Return to Summary Table.

QDMA Secondary Event Clear Register: The secondary event clear register is used to clear the status of the QSER and QER register (note that this is slightly different than the SER operation which does not clear the ER.En register). CPU write of '1' to the QSECR.En bit clears the QSER.En and QER.En register fields. CPU write of '0' has no effect..

Figure 12-139. EDMA_TPCC_QSECR_RN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES82															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES82								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-132. EDMA_TPCC_QSECR_RN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RES82	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

12.6.1.1.115 EDMA_TPCC_PARAMSET Register (Offset = 4000h)

EDMA_TPCC_PARAMSET is shown in Figure 12-140 and described in Table 12-133.

Return to Summary Table.

Bundle description is not available

Figure 12-140. EDMA_TPCC_PARAMSET Register

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-133. EDMA_TPCC_PARAMSET Register Field Descriptions

Bit	Field	Type	Reset	Description

12.6.1.1.116 EDMA_TPCC_OPT Register (Offset = 4000h) [reset = 0h]

EDMA_TPCC_OPT is shown in Figure 12-141 and described in Table 12-134.

Return to Summary Table.

Options Parameter

Figure 12-141. EDMA_TPCC_OPT Register

31	30	29	28	27	26	25	24
PRIV	RES83			PRIVID			
R-0h	R-0h			R-0h			
23	22	21	20	19	18	17	16
ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	WIMODE	RES84	TCC	
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	
15	14	13	12	11	10	9	8
TCC				TCCMODE	FWID		
R/W-0h				R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RES85				STATIC	SYNCDIM	DAM	SAM
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-134. EDMA_TPCC_OPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PRIV	R	0h	Privilege level: privilege level (supervisor vs. user) for the host/cpu/dma that programmed this PaRAM Entry. Value is set with the vbus priv value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus. PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege
30-28	RES83	R	0h	RESERVE FIELD
27-24	PRIVID	R	0h	Privilege ID: Privilege ID for the external host/cpu/dma that programmed this PaRAM Entry. This value is set with the vbus privid value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus.
23	ITCCHEN	R/W	0h	Intermediate transfer completion chaining enable: 0: Intermediate transfer complete chaining is disabled. 1: Intermediate transfer complete chaining is enabled.
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	ITCINTEN	R/W	0h	Intermediate transfer completion interrupt enable: 0: Intermediate transfer complete interrupt is disabled. 1: Intermediate transfer complete interrupt is enabled (corresponding IER[TCC] bit must be set to 1 to generate interrupt)
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled (corresponding IER[TCC] bit must be set to 1 to generate interrupt)
19	WIMODE	R/W	0h	Backward compatibility mode: 0: Normal operation 1 : WI Backwards Compatibility mode forces BCNT to be adjusted by '1' upon TR submission (0 means 1 1 means 2 ...) and forces ACNT to be treated as a word-count (left shifted by 2 by hardware to create byte cnt for TR submission)
18	RES84	R	0h	RESERVE FIELD
17-12	TCC	R/W	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER (bit CER[TCC]) for chaining or in IER (bit IER[TCC]) for interrupts.

Table 12-134. EDMA_TPCC_OPT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	TCCMODE	R/W	0h	Transfer complete code mode: Indicates the point at which a transfer is considered completed. Applies to both chaining and interrupt. 0: Normal Completion A transfer is considered completed after the transfer parameters are returned to the CC from the TC (which was returned from the peripheral). 1: Early Completion A transfer is considered completed after the CC submits a TR to the TC. CC generates completion code internally .
10-8	FWID	R/W	0h	FIFO width: Applies if either SAM or DAM is set to FIFO mode. Pass-thru to TC.
7-4	RES85	R	0h	RESERVE FIELD
3	STATIC	R/W	0h	Static Entry: 0: Entry is updated as normal 1: Entry is static Count and Address updates are not updated after TRP is submitted. Linking is not performed.
2	SYNCDIM	R/W	0h	Transfer Synchronization Dimension: 0: A-Sync Each event triggers the transfer of ACNT elements. 1: AB-Sync Each event triggers the transfer of BCNT arrays of ACNT elements
1	DAM	R/W	0h	Destination Address Mode: Destination Address Mode within an array. Pass-thru to TC. 0: INCR Dst addressing within an array increments. Dst is not a FIFO. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R/W	0h	Source Address Mode: Source Address Mode within an array. Pass-thru to TC. 0: INCR Src addressing within an array increments. Source is not a FIFO. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

12.6.1.1.117 EDMA_TPCC_SRC Register (Offset = 4004h) [reset = 0h]

EDMA_TPCC_SRC is shown in Figure 12-142 and described in Table 12-135.

Return to Summary Table.

Source Address

Figure 12-142. EDMA_TPCC_SRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-135. EDMA_TPCC_SRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SRC	R/W	0h	Source Address: The 32-bit source address parameters specify the starting byte address of the source . If SAM is set to FIFO mode then the user should program the Source address to be aligned to the value specified by the OPT.FWID field. No errors are recognized here but TC will assert error if this is not true.

12.6.1.1.118 EDMA_TPCC_ABCNT Register (Offset = 4008h) [reset = 0h]

EDMA_TPCC_ABCNT is shown in Figure 12-143 and described in Table 12-136.

Return to Summary Table.

A and B byte count

Figure 12-143. EDMA_TPCC_ABCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															
R/W-0h																R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-136. EDMA_TPCC_ABCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R/W	0h	BCNT : Count for 2nd Dimension: BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation valid values for BCNT can be anywhere between 1 and 65535. Therefore the maximum number of arrays in a frame is 65535 (64K-1 arrays). BCNT=1 means 1 array in the frame and BCNT=0 means 0 arrays in the frame. In normal mode a BCNT of '0' is considered as either a Null or Dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the OPT.WIMODE bit is set then the programmed BCNT value will be incremented by '1' before submission to TC. I.e. 0 means 1 1 means 2 2 means 3 ...
15-0	ACNT	R/W	0h	ACNT : number of bytes in 1st dimension: ACNT represents the number of bytes within the first dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65535. Therefore the maximum number of bytes in an array is 65535 bytes (64K-1 bytes). ACNT must be greater than or equal to '1' for a TR to be submitted to TC. An ACNT of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the OPT.WIMODE bit is set then the ACNT field represents a word count. The CC must internally multiply by 4 to translate the word count to a byte count prior to submission to the TC. The 2 MSBs of the 16-bit ACNT are reserved and should always be written as 'b00 by the user. If user writes a value other than 0 it will still be treated as 0 since the multiply-by-4 operation (to translate between a word count and a byte count) will drop the 2 msbits. For dummy and null transfer definition the ACNT definition will disregard the 2 msbits. I.e. a programmed ACNT value of 0x8000 in WI-mode will be treated as 0 byte transfer resulting in null or dummy operation dependent on the state of BCNT and CCNT.

12.6.1.1.119 EDMA_TPCC_DST Register (Offset = 400Ch) [reset = 0h]

EDMA_TPCC_DST is shown in Figure 12-144 and described in Table 12-137.

Return to Summary Table.

Destination Address

Figure 12-144. EDMA_TPCC_DST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-137. EDMA_TPCC_DST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DST	R/W	0h	Destination Address: The 32-bit destination address parameters specify the starting byte address of the destination. If DAM is set to FIFO mode then the user should program the Destination address to be aligned to the value specified by the OPT.FWID field. No errors are recognized here but TC will assert error if this is not true.

12.6.1.1.120 EDMA_TPCC_BIDX Register (Offset = 4010h) [reset = 0h]

EDMA_TPCC_BIDX is shown in Figure 12-145 and described in Table 12-138.

Return to Summary Table.

Register description is not available

Figure 12-145. EDMA_TPCC_BIDX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R/W-0h																R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-138. EDMA_TPCC_BIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R/W	0h	Destination 2nd Dimension Index: DBIDX is a 16-bit signed value (2's complement) used for destination address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-Sync and AB-Sync transfers.
15-0	SBIDX	R/W	0h	Source 2nd Dimension Index: SBIDX is a 16-bit signed value (2's complement) used for source address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-sync and AB-sync transfers.

12.6.1.1.121 EDMA_TPCC_LNK Register (Offset = 4014h) [reset = 0h]

EDMA_TPCC_LNK is shown in Figure 12-146 and described in Table 12-139.

Return to Summary Table.

Link and Reload parameters

Figure 12-146. EDMA_TPCC_LNK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNTRLD																LINK															
R/W-0h																R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-139. EDMA_TPCC_LNK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNTRLD	R/W	0h	BCNT Reload: BCNTRLD is a 16-bit unsigned value used to reload the BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-Sync'ed transfers. In this case the CC decrements the BCNT value by one on each TR submission. When BCNT (conceptually) reaches zero then the CC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value. For AB-synchronized transfers the CC submits the BCNT in the TR and therefore the TC is responsible to keep track of BCNT not thus BCNTRLD is a don't care field.
15-0	LINK	R/W	0h	Link Address: The CC provides a mechanism to reload the current PaRAM Entry upon its natural termination (i.e. after count fields are decremented to '0') with a new PaRAM Entry. This is called 'linking'. The 16-bit parameter LINK specifies the byte address offset in the PaRAM from which the CC loads/reloads the next PaRAM entry in the link. The CC should disregard the value in the upper 2 bits of the LINK field as well as the lower 5-bits of the LINK field. The upper two bits are ignored such that the user can program either the 'literal' byte address of the LINK parameter or the 'PaRAM base-relative' address of the link field. Therefore if the user uses the literal address with a range from 0x4000 to 0x7FFF it will be treated as a PaRAM-base-relative value of 0x0000 to 0x3FFF. The lower-5 bits are ignored and treated as 'b00000 thereby guaranteeing that all Link pointers point to a 32-byte aligned PaRAM entry. In the latter case (5-lsbs) behavior is undefined for the user (i.e. don't have to test it). In the former case (2 msbs) user should be able to take advantage of this feature (i.e. do have to test it). If a Link Update is requested to a PaRAM address that is beyond the actual range of implemented PaRAM then the Link will be treated as a Null Linkand all 0s plus 0xFFFF will be written to the current entry location. A LINK value of 0xFFFF is referred to as a NULL link which should cause the CC to write 0x0 to all entries of the current PaRAM Entry except for the LINK field which is set to 0xFFFF. The Priv/Privid/Secure state is overwritten to 0x0 when linking. MSBs and LSBS should not be masked when comparing against the 0xFFFF value. I.e. a value of 0x3FFE is a non-NULL PaRAM link field.

12.6.1.1.122 EDMA_TPCC_CIDX Register (Offset = 4018h) [reset = 0h]

EDMA_TPCC_CIDX is shown in Figure 12-147 and described in Table 12-140.

Return to Summary Table.

Register description is not available

Figure 12-147. EDMA_TPCC_CIDX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCIDX																SCIDX															
R/W-0h																R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-140. EDMA_TPCC_CIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DCIDX	R/W	0h	Destination Frame Index: DCIDX is a 16-bit signed value (2's complement) used for destination address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when DCIDX is applied the current array in an A-sync transfer is the last array in the frame while the current array in a ABsync transfer is the first array in the frame.
15-0	SCIDX	R/W	0h	Source Frame Index: SCIDX is a 16-bit signed value (2's complement) used for source address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when SCIDX is applied the current array in an A-sync transfer is the last array in the frame while the current array in a AB-sync transfer is the first array in the frame.

12.6.1.1.123 EDMA_TPCC_CCNT Register (Offset = 401Ch) [reset = 0h]

EDMA_TPCC_CCNT is shown in Figure 12-148 and described in Table 12-141.

Return to Summary Table.

C byte count

Figure 12-148. EDMA_TPCC_CCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES86																CCNT															
R-0h																R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-141. EDMA_TPCC_CCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RES86	R	0h	RESERVE FIELD
15-0	CCNT	R/W	0h	CCNT : Count for 3rd Dimension: CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT can be anywhere between 1 and 65535. Therefore the maximum number of frames in a block is 65535 (64K-1 frames). CCNT of '1' means '1' frame in the block and CCNT of '0' means '0' frames in the block. A CCNT value of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. WIMODE has no affect on CCNT operation.

12.6.1.2 EDMA_TPTC Registers

Table 12-142 lists the memory-mapped registers for the EDMA_TPTC. All register offset addresses not listed in Table 12-142 should be considered as reserved locations and the register contents should not be modified.

Table 12-142. EDMA_TPTC Registers

Offset	Acronym	Register Name	Section
0h	EDMA_TPTC_PID	EDMA_TPTC_PID	Section 12.6.1.2.1
4h	EDMA_TPTC_TCCFG	EDMA_TPTC_TCCFG	Section 12.6.1.2.2
100h	EDMA_TPTC_TCSTAT	EDMA_TPTC_TCSTAT	Section 12.6.1.2.3
104h	EDMA_TPTC_INTSTAT	EDMA_TPTC_INTSTAT	Section 12.6.1.2.4
108h	EDMA_TPTC_INTEN	EDMA_TPTC_INTEN	Section 12.6.1.2.5
10Ch	EDMA_TPTC_INTCLR	EDMA_TPTC_INTCLR	Section 12.6.1.2.6
110h	EDMA_TPTC_INTCMD	EDMA_TPTC_INTCMD	Section 12.6.1.2.7
120h	EDMA_TPTC_ERRSTAT	EDMA_TPTC_ERRSTAT	Section 12.6.1.2.8
124h	EDMA_TPTC_ERREN	EDMA_TPTC_ERREN	Section 12.6.1.2.9
128h	EDMA_TPTC_ERRCLR	EDMA_TPTC_ERRCLR	Section 12.6.1.2.10
12Ch	EDMA_TPTC_ERRDET	EDMA_TPTC_ERRDET	Section 12.6.1.2.11
130h	EDMA_TPTC_ERRCMD	EDMA_TPTC_ERRCMD	Section 12.6.1.2.12
140h	EDMA_TPTC_RDRATE	EDMA_TPTC_RDRATE	Section 12.6.1.2.13
200h	EDMA_TPTC_POPT	EDMA_TPTC_POPT	Section 12.6.1.2.14
204h	EDMA_TPTC_PSRC	EDMA_TPTC_PSRC	Section 12.6.1.2.15
208h	EDMA_TPTC_PCNT	EDMA_TPTC_PCNT	Section 12.6.1.2.16
20Ch	EDMA_TPTC_PDST	EDMA_TPTC_PDST	Section 12.6.1.2.17
210h	EDMA_TPTC_PBIDX	EDMA_TPTC_PBIDX	Section 12.6.1.2.18
214h	EDMA_TPTC_PMPPRXY	EDMA_TPTC_PMPPRXY	Section 12.6.1.2.19
240h	EDMA_TPTC_SAOPT	EDMA_TPTC_SAOPT	Section 12.6.1.2.20
244h	EDMA_TPTC_SASRC	EDMA_TPTC_SASRC	Section 12.6.1.2.21
248h	EDMA_TPTC_SACNT	EDMA_TPTC_SACNT	Section 12.6.1.2.22
250h	EDMA_TPTC_SABIDX	EDMA_TPTC_SABIDX	Section 12.6.1.2.23
254h	EDMA_TPTC_SAMPPRXY	EDMA_TPTC_SAMPPRXY	Section 12.6.1.2.24
258h	EDMA_TPTC_SACNTRLD	EDMA_TPTC_SACNTRLD	Section 12.6.1.2.25
25Ch	EDMA_TPTC_SASRCBREF	EDMA_TPTC_SASRCBREF	Section 12.6.1.2.26
260h	EDMA_TPTC_SADSTBREF	EDMA_TPTC_SADSTBREF	Section 12.6.1.2.27
280h	EDMA_TPTC_DFCNTRLD	EDMA_TPTC_DFCNTRLD	Section 12.6.1.2.28
284h	EDMA_TPTC_DFSRCBREF	EDMA_TPTC_DFSRCBREF	Section 12.6.1.2.29
300h	EDMA_TPTC_DFOPT	EDMA_TPTC_DFOPT	Section 12.6.1.2.30
304h	EDMA_TPTC_DFSRC	EDMA_TPTC_DFSRC	Section 12.6.1.2.31
308h	EDMA_TPTC_DFCNT	EDMA_TPTC_DFCNT	Section 12.6.1.2.32
30Ch	EDMA_TPTC_DFDST	EDMA_TPTC_DFDST	Section 12.6.1.2.33
310h	EDMA_TPTC_DFBIDX	EDMA_TPTC_DFBIDX	Section 12.6.1.2.34
314h	EDMA_TPTC_DFMPPRXY	EDMA_TPTC_DFMPPRXY	Section 12.6.1.2.35

12.6.1.2.1 EDMA_TPTC_PID Register (Offset = 0h) [reset = 4000AB00h]

EDMA_TPTC_PID is shown in Figure 12-149 and described in Table 12-143.

Return to Summary Table.

Peripheral ID Register

Figure 12-149. EDMA_TPTC_PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RES1		FUNC											
R-1h		R-0h		R-0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-15h					R-3h			R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-143. EDMA_TPTC_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Scheme: Used to distinguish between old ID scheme and current. Spare bit to encode future schemes EDMA uses 'new scheme' indicated with value of 0x1.
29-28	RES1	R	0h	RESERVE FIELD
27-16	FUNC	R	0h	Function indicates a software compatible module family.
15-11	RTL	R	15h	RTL Version
10-8	MAJOR	R	3h	Major Revision
7-6	CUSTOM	R	0h	Custom revision field: Not used on this version of EDMA.
5-0	MINOR	R	0h	Minor Revision

12.6.1.2.2 EDMA_TPTC_TCCFG Register (Offset = 4h) [reset = 122h]

EDMA_TPTC_TCCFG is shown in Figure 12-150 and described in Table 12-144.

Return to Summary Table.

TC Configuration Register

Figure 12-150. EDMA_TPTC_TCCFG Register

31	30	29	28	27	26	25	24
RES2							
R-0h							
23	22	21	20	19	18	17	16
RES2							
R-0h							
15	14	13	12	11	10	9	8
RES2						DREGDEPTH	
R-0h						R-1h	
7	6	5	4	3	2	1	0
RES3		BUSWIDTH		RES4		FIFOSIZE	
R-0h		R-2h		R-0h		R-2h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-144. EDMA_TPTC_TCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RES2	R	0h	RESERVE FIELD
9-8	DREGDEPTH	R	1h	Dst Register FIFO Depth Parameterization
7-6	RES3	R	0h	RESERVE FIELD
5-4	BUSWIDTH	R	2h	Bus Width Parameterization
3	RES4	R	0h	RESERVE FIELD
2-0	FIFOSIZE	R	2h	Fifo Size Parameterization

12.6.1.2.3 EDMA_TPTC_TCSTAT Register (Offset = 100h) [reset = 100h]

EDMA_TPTC_TCSTAT is shown in Figure 12-151 and described in Table 12-145.

Return to Summary Table.

TC Status Register

Figure 12-151. EDMA_TPTC_TCSTAT Register

31	30	29	28	27	26	25	24
RES5							
R-0h							
23	22	21	20	19	18	17	16
RES5							
R-0h							
15	14	13	12	11	10	9	8
RES5			DFSTRTPTR		RES6		ACTV
R-0h			R-0h		R-0h		R-1h
7	6	5	4	3	2	1	0
RES7	DSTACTV			RES8	WSACTV	SRCACTV	PROGBUSY
R-0h	R-0h			R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-145. EDMA_TPTC_TCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES5	R	0h	RESERVE FIELD
13-12	DFSTRTPTR	R	0h	Dst FIFO Start Pointer Represents the offset to the head entry of Dst Register FIFO in units of entries. Legal values = 0x0 to 0x3
11-9	RES6	R	0h	RESERVE FIELD
8	ACTV	R	1h	Channel Active Channel Active is a logical-OR of each of the BUSY/ACTV signals. The ACTV bit must remain high through the life of a TR. ACTV = 0 : Channel is idle. ACTV = 1 : Channel is busy.
7	RES7	R	0h	RESERVE FIELD
6-4	DSTACTV	R	0h	Destination Active State Specifies the number of TRs that are resident in the Dst Register FIFO at a given instant. Legal values are constrained by the DSTREGDEPTH parameter.
3	RES8	R	0h	RESERVE FIELD
2	WSACTV	R	0h	Write Status Active WSACTV = 0 : Write status is not pending. Write status has been received for all previously issued write commands. WSACTV = 1 : Write Status is pending. Write status has not been received for all previously issued write commands.
1	SRCACTV	R	0h	Source Active State SRCACTV = 0 : Source Active set is idle. Any TR written to Prog Set will immediately transition to Source Active set as long as the Dst FIFO Set is not full (DSTFULL == 1). SRCACTV = 1 : Source Active set is busy either performing read transfers or waiting to perform read transfers for current Transfer Request.
0	PROGBUSY	R	0h	Program Register Set Busy PROGBUSY = 0 : Prog set idle and is available for programming. PROGBUSY = 1 : Prog set busy. User should poll for PROGBUSY equal to '0' prior to re-programming the Program Register set.

12.6.1.2.4 EDMA_TPTC_INTSTAT Register (Offset = 104h) [reset = 0h]

EDMA_TPTC_INTSTAT is shown in Figure 12-152 and described in Table 12-146.

Return to Summary Table.

Interrupt Status Register

Figure 12-152. EDMA_TPTC_INTSTAT Register

31	30	29	28	27	26	25	24
RES9							
R-0h							
23	22	21	20	19	18	17	16
RES9							
R-0h							
15	14	13	12	11	10	9	8
RES9							
R-0h							
7	6	5	4	3	2	1	0
RES9						TRDONE	PROGEMPTY
R-0h						R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-146. EDMA_TPTC_INTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES9	R	0h	RESERVE FIELD
1	TRDONE	R	0h	TR Done Event Status: TRDONE = 0 : Condition not detected. TRDONE = 1 : Set when TC has completed a Transfer Request. TRDONE should be set when the write status is returned for the final write of a TR. Cleared when user writes '1' to INTCLR.TRDONE register bit.
0	PROGEMPTY	R	0h	Program Set Empty Event Status: PROGEMPTY = 0 : Condition not detected. PROGEMPTY = 1 : Set when Program Register set transitions to empty state. Cleared when user writes '1' to INTCLR.PROGEMPTY register bit.

12.6.1.2.5 EDMA_TPTC_INTEN Register (Offset = 108h) [reset = 0h]

EDMA_TPTC_INTEN is shown in Figure 12-153 and described in Table 12-147.

Return to Summary Table.

Interrupt Enable Register

Figure 12-153. EDMA_TPTC_INTEN Register

31	30	29	28	27	26	25	24
RES10							
R-0h							
23	22	21	20	19	18	17	16
RES10							
R-0h							
15	14	13	12	11	10	9	8
RES10							
R-0h							
7	6	5	4	3	2	1	0
RES10						TRDONE	PROGEMPTY
R-0h						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-147. EDMA_TPTC_INTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES10	R	0h	RESERVE FIELD
1	TRDONE	R/W	0h	TR Done Event Enable: INTEN.TRDONE = 0 : TRDONE Event is disabled. INTEN.TRDONE = 1 : TRDONE Event is enabled and contributes to interrupt generation
0	PROGEMPTY	R/W	0h	Program Set Empty Event Enable: INTEN.PROGEMPTY = 0 : PROGEMPTY Event is disabled. INTEN.PROGEMPTY = 1 : PROGEMPTY Event is enabled and contributes to interrupt generation

12.6.1.2.6 EDMA_TPTC_INTCLR Register (Offset = 10Ch) [reset = 0h]

EDMA_TPTC_INTCLR is shown in Figure 12-154 and described in Table 12-148.

Return to Summary Table.

Interrupt Clear Register

Figure 12-154. EDMA_TPTC_INTCLR Register

31	30	29	28	27	26	25	24
RES11							
R-0h							
23	22	21	20	19	18	17	16
RES11							
R-0h							
15	14	13	12	11	10	9	8
RES11							
R-0h							
7	6	5	4	3	2	1	0
RES11						TRDONE	PROGEMPTY
R-0h						W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-148. EDMA_TPTC_INTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES11	R	0h	RESERVE FIELD
1	TRDONE	W	0h	TR Done Event Clear: INTCLR.TRDONE = 0 : Writes of '0' have no effect.. INTCLR.TRDONE = 1 : Write of '1' clears INTSTAT.TRDONE bit
0	PROGEMPTY	W	0h	Program Set Empty Event Clear: INTCLR.PROGEMPTY = 0 : Writes of '0' have no effect.. INTCLR.PROGEMPTY = 1 : Write of '1' clears INTSTAT.PROGEMPTY bit

12.6.1.2.7 EDMA_TPTC_INTCMD Register (Offset = 110h) [reset = 0h]

EDMA_TPTC_INTCMD is shown in Figure 12-155 and described in Table 12-149.

Return to Summary Table.

Interrupt Command Register

Figure 12-155. EDMA_TPTC_INTCMD Register

31	30	29	28	27	26	25	24
RES12							
R-0h							
23	22	21	20	19	18	17	16
RES12							
R-0h							
15	14	13	12	11	10	9	8
RES12							
R-0h							
7	6	5	4	3	2	1	0
RES12						SET	EVAL
R-0h						W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-149. EDMA_TPTC_INTCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES12	R	0h	RESERVE FIELD
1	SET	W	0h	Set TPTC interrupt: Write of '1' to SET causes TPTC interrupt to be pulsed unconditionally. Writes of '0' have no affect.
0	EVAL	W	0h	Evaluate state of TPTC interrupt Write of '1' to EVAL causes TPTC interrupt to be pulsed if any of the INTSTAT bits are set to '1'. Writes of '0' have no affect.

12.6.1.2.8 EDMA_TPTC_ERRSTAT Register (Offset = 120h) [reset = 0h]

EDMA_TPTC_ERRSTAT is shown in Figure 12-156 and described in Table 12-150.

Return to Summary Table.

Error Status Register

Figure 12-156. EDMA_TPTC_ERRSTAT Register

31	30	29	28	27	26	25	24
RES13							
R-0h							
23	22	21	20	19	18	17	16
RES13							
R-0h							
15	14	13	12	11	10	9	8
RES13							
R-0h							
7	6	5	4	3	2	1	0
RES13				MMRAERR	TRERR	RES14	BUSERR
R-0h				R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-150. EDMA_TPTC_ERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RES13	R	0h	RESERVE FIELD
3	MMRAERR	R	0h	MMR Address Error: MMRAERR = 0 : Condition not detected. MMRAERR = 1 : User attempted to read or write to invalid address configuration memory map. (Is only be set for non-emulation accesses). No additional error information is recorded.
2	TRERR	R	0h	TR Error: TR detected that violates FIFO Mode transfer (SAM or DAM is '1') alignment rules or has ACNT or BCNT == 0. No additional error information is recorded.
1	RES14	R	0h	RESERVE FIELD
0	BUSERR	R	0h	Bus Error Event: BUSERR = 0: Condition not detected. BUSERR = 1: TC has detected an error code on the write response bus or read response bus. Error information is stored in Error Details Register (ERRDET).

12.6.1.2.9 EDMA_TPTC_ERREN Register (Offset = 124h) [reset = 0h]

EDMA_TPTC_ERREN is shown in Figure 12-157 and described in Table 12-151.

Return to Summary Table.

Error Enable Register

Figure 12-157. EDMA_TPTC_ERREN Register

31	30	29	28	27	26	25	24
RES15							
R-0h							
23	22	21	20	19	18	17	16
RES15							
R-0h							
15	14	13	12	11	10	9	8
RES15							
R-0h							
7	6	5	4	3	2	1	0
RES15				MMRAERR	TRERR	RES16	BUSERR
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-151. EDMA_TPTC_ERREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RES15	R	0h	RESERVE FIELD
3	MMRAERR	R/W	0h	Interrupt enable for ERRSTAT.MMRAERR: ERREN.MMRAERR = 0 : BUSERR is disabled. ERREN.MMRAERR = 1 : MMRAERR is enabled and contributes to the TPTC error interrupt generation.
2	TRERR	R/W	0h	Interrupt enable for ERRSTAT.TRERR: ERREN.TRERR = 0 : BUSERR is disabled. ERREN.TRERR = 1 : TRERR is enabled and contributes to the TPTC error interrupt generation.
1	RES16	R	0h	RESERVE FIELD
0	BUSERR	R/W	0h	Interrupt enable for ERRSTAT.BUSERR: ERREN.BUSERR = 0 : BUSERR is disabled. ERREN.BUSERR = 1 : BUSERR is enabled and contributes to the TPTC error interrupt generation.

12.6.1.2.10 EDMA_TPTC_ERRCLR Register (Offset = 128h) [reset = 0h]

EDMA_TPTC_ERRCLR is shown in Figure 12-158 and described in Table 12-152.

Return to Summary Table.

Error Clear Register

Figure 12-158. EDMA_TPTC_ERRCLR Register

31	30	29	28	27	26	25	24
RES17							
R-0h							
23	22	21	20	19	18	17	16
RES17							
R-0h							
15	14	13	12	11	10	9	8
RES17							
R-0h							
7	6	5	4	3	2	1	0
RES17				MMRAERR	TRERR	RES18	BUSERR
R-0h				W-0h	W-0h	R-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-152. EDMA_TPTC_ERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RES17	R	0h	RESERVE FIELD
3	MMRAERR	W	0h	Interrupt clear for ERRSTAT.MMRAERR: ERRCLR.MMRAERR = 0 : Writes of '0' have no effect.. ERRCLR.MMRAERR = 1 : Write of '1' clears ERRSTAT.MMRAERR bit. Write of '1' to ERRCLR.MMRAERR does not clear the ERRDET register.
2	TRERR	W	0h	Interrupt clear for ERRSTAT.TRERR: ERRCLR.TRERR = 0 : Writes of '0' have no effect.. ERRCLR.TRERR = 1 : Write of '1' clears ERRSTAT.TRERR bit. Write of '1' to ERRCLR.TRERR does not clear the ERRDET register.
1	RES18	R	0h	RESERVE FIELD
0	BUSERR	W	0h	Interrupt clear for ERRSTAT.BUSERR: ERRCLR.BUSERR = 0 : Writes of '0' have no effect.. ERRCLR.BUSERR = 1 : Write of '1' clears ERRSTAT.BUSERR bit. Write of '1' to ERRCLR.BUSERR clears the ERRDET register.

12.6.1.2.11 EDMA_TPTC_ERRDET Register (Offset = 12Ch) [reset = 0h]

EDMA_TPTC_ERRDET is shown in Figure 12-159 and described in Table 12-153.

Return to Summary Table.

Error Details Register

Figure 12-159. EDMA_TPTC_ERRDET Register

31	30	29	28	27	26	25	24
RES19							
R-0h							
23	22	21	20	19	18	17	16
RES19						TCCHEN	TCINTEN
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RES20		TCC					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RES21				STAT			
R-0h				R-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-153. EDMA_TPTC_ERRDET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RES19	R	0h	RESERVE FIELD
17	TCCHEN	R	0h	Contains the OPT.TCCHEN value programmed by the user for the Read or Write transaction that resulted in an error.
16	TCINTEN	R	0h	Contains the OPT.TCINTEN value programmed by the user for the Read or Write transaction that resulted in an error.
15-14	RES20	R	0h	RESERVE FIELD
13-8	TCC	R	0h	Transfer Complete Code: Contains the OPT.TCC value programmed by the user for the Read or Write transaction that resulted in an error.
7-4	RES21	R	0h	RESERVE FIELD
3-0	STAT	R	0h	Transaction Status: Stores the non-zero status/error code that was detected on the read status or write status bus. MS-bit effectively serves as the read vs. write error code. If read status and write status are returned on the same cycle then the TC chooses non-zero version. If both are non-zero then write status is treated as higher priority. Encoding of errors matches the CBA spec and is summarized here: 0h = No error 1h - 7h = Read error 8h - Fh = Write error

12.6.1.2.12 EDMA_TPTC_ERRCMD Register (Offset = 130h) [reset = 0h]

EDMA_TPTC_ERRCMD is shown in Figure 12-160 and described in Table 12-154.

Return to Summary Table.

Error Command Register

Figure 12-160. EDMA_TPTC_ERRCMD Register

31	30	29	28	27	26	25	24
RES22							
R-0h							
23	22	21	20	19	18	17	16
RES22							
R-0h							
15	14	13	12	11	10	9	8
RES22							
R-0h							
7	6	5	4	3	2	1	0
RES22						SET	EVAL
R-0h						W-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-154. EDMA_TPTC_ERRCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES22	R	0h	RESERVE FIELD
1	SET	W	0h	Set TPTC error interrupt: Write of '1' to SET causes TPTC error interrupt to be pulsed unconditionally. Writes of '0' have no affect.
0	EVAL	W	0h	Evaluate state of TPTC error interrupt Write of '1' to EVAL causes TPTC error interrupt to be pulsed if any of the ERRSTAT bits are set to '1'. Writes of '0' have no affect.

12.6.1.2.13 EDMA_TPTC_RDRATE Register (Offset = 140h) [reset = 0h]

EDMA_TPTC_RDRATE is shown in Figure 12-161 and described in Table 12-155.

Return to Summary Table.

Read Rate Register

Figure 12-161. EDMA_TPTC_RDRATE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES23															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES23													RDRATE		
R-0h													R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-155. EDMA_TPTC_RDRATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RES23	R	0h	RESERVE FIELD
2-0	RDRATE	R/W	0h	Read Rate Control: Controls the number of cycles between read commands. This is a global setting that applies to all TRs for this TC.

12.6.1.2.14 EDMA_TPTC_POPT Register (Offset = 200h) [reset = 0h]

EDMA_TPTC_POPT is shown in Figure 12-162 and described in Table 12-156.

Return to Summary Table.

Prog Set Options

Figure 12-162. EDMA_TPTC_POPT Register

31	30	29	28	27	26	25	24
RES24							
R-0h							
23	22	21	20	19	18	17	16
RES24	TCCHEN	RES25	TCINTEN	RES26		TCC	
R-0h	R/W-0h	R-0h	R/W-0h	R-0h		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RES27	FWID		
R/W-0h				R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RES28	PRI			RES29		DAM	SAM
R-0h	R/W-0h			R-0h		R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-156. EDMA_TPTC_POPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RES24	R	0h	RESERVE FIELD
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RES25	R	0h	RESERVE FIELD
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RES26	R	0h	RESERVE FIELD
17-12	TCC	R/W	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or IPR of the TPCC module.
11	RES27	R	0h	RESERVE FIELD
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RES28	R	0h	RESERVE FIELD
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RES29	R	0h	RESERVE FIELD
1	DAM	R/W	0h	Destination Address Mode within an array: 0: INCR Dst addressing within an array increments. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R/W	0h	Source Address Mode within an array: 0: INCR Src addressing within an array increments. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

12.6.1.2.15 EDMA_TPTC_PSRC Register (Offset = 204h) [reset = 0h]

EDMA_TPTC_PSRC is shown in Figure 12-163 and described in Table 12-157.

Return to Summary Table.

Prog Set Src Address

Figure 12-163. EDMA_TPTC_PSRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-157. EDMA_TPTC_PSRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDR	R/W	0h	Source address for Program Register Set

12.6.1.2.16 EDMA_TPTC_PCNT Register (Offset = 208h) [reset = 0h]

EDMA_TPTC_PCNT is shown in Figure 12-164 and described in Table 12-158.

Return to Summary Table.

Prog Set Count

Figure 12-164. EDMA_TPTC_PCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															
R/W-0h																R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-158. EDMA_TPTC_PCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R/W	0h	B-Dimension count. Number of arrays to be transferred where each array is ACNT in length.
15-0	ACNT	R/W	0h	A-Dimension count. Number of bytes to be transferred in first dimension.

12.6.1.2.17 EDMA_TPTC_PDST Register (Offset = 20Ch) [reset = 0h]

EDMA_TPTC_PDST is shown in Figure 12-165 and described in Table 12-159.

Return to Summary Table.

Prog Set Dst Address

Figure 12-165. EDMA_TPTC_PDST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-159. EDMA_TPTC_PDST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R/W	0h	Destination address for Program Register Set

12.6.1.2.18 EDMA_TPTC_PBIDX Register (Offset = 210h) [reset = 0h]

EDMA_TPTC_PBIDX is shown in Figure 12-166 and described in Table 12-160.

Return to Summary Table.

Prog Set B-Dim Idx

Figure 12-166. EDMA_TPTC_PBIDX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R/W-0h																R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-160. EDMA_TPTC_PBIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R/W	0h	Dest B-Idx for Program Register Set: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used regardless of whether DAM is Increment or FIFO mode.
15-0	SBIDX	R/W	0h	Source B-Idx for Program Register Set: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is Increment or FIFO mode.

12.6.1.2.19 EDMA_TPTC_PMPPRXY Register (Offset = 214h) [reset = 0h]

EDMA_TPTC_PMPPRXY is shown in Figure 12-167 and described in Table 12-161.

Return to Summary Table.

Prog Set Mem Protect Proxy

Figure 12-167. EDMA_TPTC_PMPPRXY Register

31	30	29	28	27	26	25	24
RES30							
R-0h							
23	22	21	20	19	18	17	16
RES30							
R-0h							
15	14	13	12	11	10	9	8
RES30						SECURE	PRIV
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RES31				PRIVID			
R-0h				R-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-161. EDMA_TPTC_PMPPRXY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RES30	R	0h	RESERVE FIELD
9	SECURE	R	0h	Secure Level: SECURE = 0 : Non-secure access SECURE = 1 : Secure access PMPPRXY.SECURE is always updated with the value from the configuration bus secure field on any/every write to Program Set BIDX Register (trigger register). The SECURE value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The secure attribute is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection and security checks based on the SECURE level of the external host that sets up the DMA transaction.
8	PRIV	R	0h	Privilege Level: PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege PMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the PRIV of the external host that sets up the DMA transaction.
7-4	RES31	R	0h	RESERVE FIELD
3-0	PRIVID	R	0h	Privilege ID: PMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the privid of the external host that sets up the DMA transaction.

12.6.1.2.20 EDMA_TPTC_SAOPT Register (Offset = 240h) [reset = 0h]

EDMA_TPTC_SAOPT is shown in Figure 12-168 and described in Table 12-162.

Return to Summary Table.

Src Actv Set Options

Figure 12-168. EDMA_TPTC_SAOPT Register

31	30	29	28	27	26	25	24
RES32							
R-0h							
23	22	21	20	19	18	17	16
RES32	TCCHEN	RES33	TCINTEN	RES34		TCC	
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	
15	14	13	12	11	10	9	8
TCC				RES35	FWID		
R-0h				R-0h	R-0h		
7	6	5	4	3	2	1	0
RES36	PRI			RES37		DAM	SAM
R-0h	R-0h			R-0h		R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-162. EDMA_TPTC_SAOPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RES32	R	0h	RESERVE FIELD
22	TCCHEN	R	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RES33	R	0h	RESERVE FIELD
20	TCINTEN	R	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RES34	R	0h	RESERVE FIELD
17-12	TCC	R	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or IPR of the TPCC module.
11	RES35	R	0h	RESERVE FIELD
10-8	FWID	R	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RES36	R	0h	RESERVE FIELD
6-4	PRI	R	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RES37	R	0h	RESERVE FIELD
1	DAM	R	0h	Destination Address Mode within an array: 0: INCR Dst addressing within an array increments. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R	0h	Source Address Mode within an array: 0: INCR Src addressing within an array increments. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

12.6.1.2.21 EDMA_TPTC_SASRC Register (Offset = 244h) [reset = 0h]

EDMA_TPTC_SASRC is shown in Figure 12-169 and described in Table 12-163.

Return to Summary Table.

Src Actv Set Src Address

Figure 12-169. EDMA_TPTC_SASRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-163. EDMA_TPTC_SASRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDR	R	0h	Source address for Source Active Register Set: Initial value is copied from PSRC.SADDR. TC updates value according to source addressing mode (OPT.SAM) and/or source index value (BIDX.SBIDX) after each read command is issued. When a TR is complete the final value should be the address of the last read command issued.

12.6.1.2.22 EDMA_TPTC_SACNT Register (Offset = 248h) [reset = 0h]

EDMA_TPTC_SACNT is shown in Figure 12-170 and described in Table 12-164.

Return to Summary Table.

Src Actv Set Count

Figure 12-170. EDMA_TPTC_SACNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															
R-0h																R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-164. EDMA_TPTC_SACNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R	0h	B-Dimension count: Number of arrays to be transferred where each array is ACNT in length. Count Remaining for Src Active Register Set. Represents the amount of data remaining to be read. Initial value is copied from PCNT. TC decrements ACNT and BCNT as necessary after each read command is issued. Final value should be 0 when TR is complete.
15-0	ACNT	R	0h	A-Dimension count: Number of bytes to be transferred in first dimension. Count Remaining for Src Active Register Set. Represents the amount of data remaining to be read. Initial value is copied from PCNT. TC decrements ACNT and BCNT as necessary after each read command is issued. Final value should be 0 when TR is complete.

12.6.1.2.23 EDMA_TPTC_SABIDX Register (Offset = 250h) [reset = 0h]

EDMA_TPTC_SABIDX is shown in Figure 12-171 and described in Table 12-165.

Return to Summary Table.

Src Actv Set B-Dim Idx

Figure 12-171. EDMA_TPTC_SABIDX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R-0h																R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-165. EDMA_TPTC_SABIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	Dest B-Idx for Source Active Register Set. Value copied from PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used regardless of whether DAM is Increment or FIFO mode.
15-0	SBIDX	R	0h	Source B-Idx for Source Active Register Set. Value copied from PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is Increment or FIFO mode.

12.6.1.2.24 EDMA_TPTC_SAMPPRXY Register (Offset = 254h) [reset = 0h]

EDMA_TPTC_SAMPPRXY is shown in Figure 12-172 and described in Table 12-166.

Return to Summary Table.

Src Actv Set Mem Protect Proxy

Figure 12-172. EDMA_TPTC_SAMPPRXY Register

31	30	29	28	27	26	25	24
RES38							
R-0h							
23	22	21	20	19	18	17	16
RES38							
R-0h							
15	14	13	12	11	10	9	8
RES38						SECURE	PRIV
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RES39				PRIVID			
R-0h				R-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-166. EDMA_TPTC_SAMPPRXY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RES38	R	0h	RESERVE FIELD
9	SECURE	R	0h	Secure Level: SECURE = 0 : Non-secure access SECURE = 1 : Secure access SAMPPRXY.SECURE is always updated with the value from the configuration bus secure field on any/every write to Program Set BIDX Register (trigger register). The SECURE value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The secure attribute is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection and security checks based on the SECURE level of the external host that sets up the DMA transaction.
8	PRIV	R	0h	Privilege Level: PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege SAMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the PRIV of the external host that sets up the DMA transaction.
7-4	RES39	R	0h	RESERVE FIELD
3-0	PRIVID	R	0h	Privilege ID: SAMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the privid of the external host that sets up the DMA transaction.

12.6.1.2.25 EDMA_TPTC_SACNTRLD Register (Offset = 258h) [reset = 0h]

EDMA_TPTC_SACNTRLD is shown in Figure 12-173 and described in Table 12-167.

Return to Summary Table.

Src Actv Set Cnt Reload

Figure 12-173. EDMA_TPTC_SACNTRLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES40																ACNTRLD															
R-0h																R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-167. EDMA_TPTC_SACNTRLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RES40	R	0h	RESERVE FIELD
15-0	ACNTRLD	R	0h	A-Cnt Reload value for Source Active Register set. Value copied from PCNT.ACNT: Represents the originally programmed value of ACNT. The Reload value is used to reinitialize ACNT after each array is serviced (i.e. ACNT decrements to 0). by the Src offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT bytes)

12.6.1.2.26 EDMA_TPTC_SASRCBREF Register (Offset = 25Ch) [reset = 0h]

EDMA_TPTC_SASRCBREF is shown in Figure 12-174 and described in Table 12-168.

Return to Summary Table.

Src Actv Set Src Addr A-Reference

Figure 12-174. EDMA_TPTC_SASRCBREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-168. EDMA_TPTC_SASRCBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDRBREF	R	0h	Source address reference for Source Active Register Set: Represents the starting address for the array currently being read. The next array's starting address is calculated as the 'reference address' plus the 'source b-idx' value.

12.6.1.2.27 EDMA_TPTC_SADSTBREF Register (Offset = 260h) [reset = 0h]

EDMA_TPTC_SADSTBREF is shown in Figure 12-175 and described in Table 12-169.

Return to Summary Table.

rsvd return 0x0 w/o AERROR

Figure 12-175. EDMA_TPTC_SADSTBREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDRBREF																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-169. EDMA_TPTC_SADSTBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDRBREF	R	0h	Dst address reference is not applicable for Src Active Register Set. Reads return 0x0.

12.6.1.2.28 EDMA_TPTC_DFCNTRLD Register (Offset = 280h) [reset = 0h]

EDMA_TPTC_DFCNTRLD is shown in Figure 12-176 and described in Table 12-170.

Return to Summary Table.

Dst FIFO Set Cnt Reload

Figure 12-176. EDMA_TPTC_DFCNTRLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES41																ACNTRLD															
R-0h																R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-170. EDMA_TPTC_DFCNTRLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RES41	R	0h	RESERVE FIELD
15-0	ACNTRLD	R	0h	A-Cnt Reload value for Destination FIFO Register set. Value copied from PCNT.ACNT: Represents the originally programmed value of ACNT. The Reload value is used to reinitialize ACNT after each array is serviced (i.e. ACNT decrements to 0). by the Src offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT bytes)

12.6.1.2.29 EDMA_TPTC_DFSRCBREF Register (Offset = 284h) [reset = 0h]

EDMA_TPTC_DFSRCBREF is shown in Figure 12-177 and described in Table 12-171.

Return to Summary Table.

rsvd return 0x0 w/o AERROR

Figure 12-177. EDMA_TPTC_DFSRCBREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-171. EDMA_TPTC_DFSRCBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDRBREF	R	0h	Source address reference is not applicable for Dst FIFO Register Set. Reads return 0x0.

12.6.1.2.30 EDMA_TPTC_DFOPT Register (Offset = 300h) [reset = 0h]

EDMA_TPTC_DFOPT is shown in Figure 12-178 and described in Table 12-172.

Return to Summary Table.

Dst FIFO Set Options

Figure 12-178. EDMA_TPTC_DFOPT Register

31	30	29	28	27	26	25	24
RES42							
R-0h							
23	22	21	20	19	18	17	16
RES42	TCCHEN	RES43	TCINTEN	RES44		TCC	
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	
15	14	13	12	11	10	9	8
TCC				RES45	FWID		
R-0h				R-0h	R-0h		
7	6	5	4	3	2	1	0
RES46	PRI			RES47		DAM	SAM
R-0h	R-0h			R-0h		R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-172. EDMA_TPTC_DFOPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RES42	R	0h	RESERVE FIELD
22	TCCHEN	R	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RES43	R	0h	RESERVE FIELD
20	TCINTEN	R	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RES44	R	0h	RESERVE FIELD
17-12	TCC	R	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or IPR of the TPCC module.
11	RES45	R	0h	RESERVE FIELD
10-8	FWID	R	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RES46	R	0h	RESERVE FIELD
6-4	PRI	R	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RES47	R	0h	RESERVE FIELD
1	DAM	R	0h	Destination Address Mode within an array: 0: INCR Dst addressing within an array increments. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R	0h	Source Address Mode within an array: 0: INCR Src addressing within an array increments. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

12.6.1.2.31 EDMA_TPTC_DFSRC Register (Offset = 304h) [reset = 0h]

EDMA_TPTC_DFSRC is shown in Figure 12-179 and described in Table 12-173.

Return to Summary Table.

rsvd return 0x0 w/o AERROR

Figure 12-179. EDMA_TPTC_DFSRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-173. EDMA_TPTC_DFSRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDR	R	0h	Source address is not applicable for Dst FIFO Register Set: Reads return 0x0.

12.6.1.2.32 EDMA_TPTC_DFCNT Register (Offset = 308h) [reset = 0h]

EDMA_TPTC_DFCNT is shown in Figure 12-180 and described in Table 12-174.

Return to Summary Table.

Dst FIFO Set Count

Figure 12-180. EDMA_TPTC_DFCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															
R-0h																R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-174. EDMA_TPTC_DFCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BCNT	R	0h	B-Count Remaining for Dst Register Set: Number of arrays to be transferred where each array is ACNT in length. Represents the amount of data remaining to be written. Initial value is copied from PCNT. TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.
15-0	ACNT	R	0h	A-Count Remaining for Dst Register Set: Number of bytes to be transferred in first dimension. Represents the amount of data remaining to be written. Initial value is copied from PCNT. TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.

12.6.1.2.33 EDMA_TPTC_DFDST Register (Offset = 30Ch) [reset = 0h]

EDMA_TPTC_DFDST is shown in Figure 12-181 and described in Table 12-175.

Return to Summary Table.

Dst FIFO Set Dst Address

Figure 12-181. EDMA_TPTC_DFDST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-175. EDMA_TPTC_DFDST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	Destination address for Dst FIFO Register Set: Initial value is copied from PDST.DADDR. TC updates value according to destination addressing mode (OPT.SAM) and/or dest index value (BIDX.DBIDX) after each write command is issued. When a TR is complete the final value should be the address of the last write command issued.

12.6.1.2.34 EDMA_TPTC_DFBIDX Register (Offset = 310h) [reset = 0h]

EDMA_TPTC_DFBIDX is shown in Figure 12-182 and described in Table 12-176.

Return to Summary Table.

Dst FIFO Set B-Dim Idx

Figure 12-182. EDMA_TPTC_DFBIDX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R-0h																R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-176. EDMA_TPTC_DFBIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	Dest B-Idx for Dest FIFO Register Set. Value copied from PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used regardless of whether DAM is Increment or FIFO mode.
15-0	SBIDX	R	0h	Dest B-Idx for Dest FIFO Register Set. Value copied from PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used regardless of whether SAM is Increment or FIFO mode.

12.6.1.2.35 EDMA_TPTC_DFMPPRXY Register (Offset = 314h) [reset = 0h]

EDMA_TPTC_DFMPPRXY is shown in Figure 12-183 and described in Table 12-177.

Return to Summary Table.

Dst FIFO Mem Protect Proxy

Figure 12-183. EDMA_TPTC_DFMPPRXY Register

31	30	29	28	27	26	25	24
RES48							
R-0h							
23	22	21	20	19	18	17	16
RES48							
R-0h							
15	14	13	12	11	10	9	8
RES48						SECURE	PRIV
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RES49				PRIVID			
R-0h				R-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-177. EDMA_TPTC_DFMPPRXY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RES48	R	0h	RESERVE FIELD
9	SECURE	R	0h	Secure Level: SECURE = 0 : Non-secure access SECURE = 1 : Secure access DFMPPRXY0.SECURE is always updated with the value from the configuration bus secure field on any/every write to Program Set BIDX Register (trigger register). The SECURE value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The secure attribute is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection and security checks based on the SECURE level of the external host that sets up the DMA transaction.
8	PRIV	R	0h	Privilege Level: PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege DFMPPRXY0.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the PRIV of the external host that sets up the DMA transaction.
7-4	RES49	R	0h	RESERVE FIELD
3-0	PRIVID	R	0h	Privilege ID: DFMPPRXY0.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the privid of the external host that sets up the DMA transaction.

ADC Buffer

The ADC buffer is on-chip memory arranged as a ping-pong buffer, with ECC support for each ping and pong memory. The raw ADC output data from RADAR-SS is stored on this memory, to be consumed by the DSP with the 16xx/18xx, or by the hardware FFT accelerator with the 14xx, or by the DSP or hardware FFT accelerator with the 68xx for the post processing. The buffer size details are:

Device	14xx	16xx/68xx/18xx
Size in KB	16	32

For the application software, the ADC buffer (either ping or pong) is seen as a single memory at the base address, as specified in , and size as mentioned above. For the register details, refer to .

Topic **Page**

13.1 Functional Description **1846**

13.1 Functional Description

Figure 13-2 shows the block diagram of the ADC buffer scheme. The three data input sources to the ADC buffer are:

- Supported in the 14xx and 16xx/18xx/68xx:
  - Raw ADC output data from the digital front end (DFE)
  - Ramp pattern data from the test pattern generator
- Supported in the 16xx/18xx/68xx only:
  - HIL data from the DMM interface, as shown in Figure 13-1 and Figure 13-2.

Figure 13-1. ADC Buffer Block Diagram for the 14xx

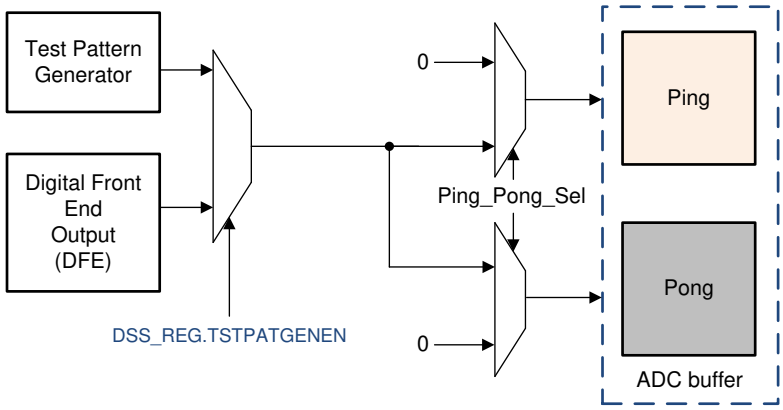
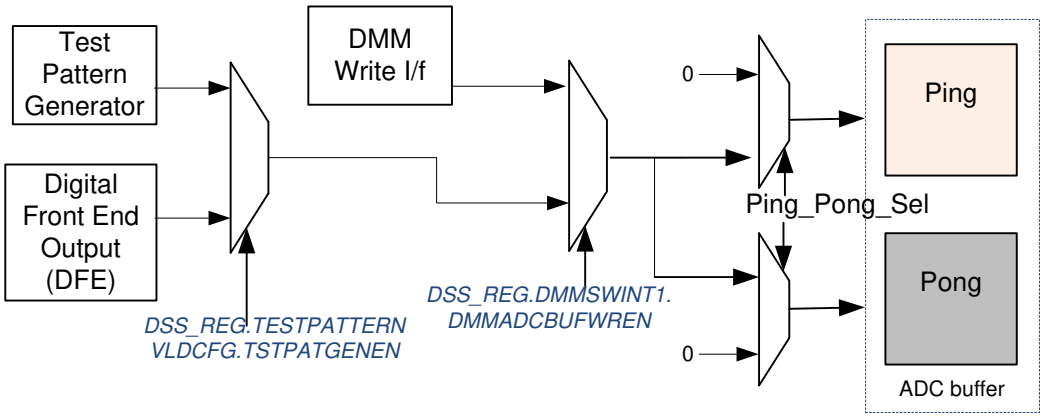


Figure 13-2. ADC Buffer Block Diagram for the 16xx/18xx/68xx



13.1.1 DFE Data Write Operation

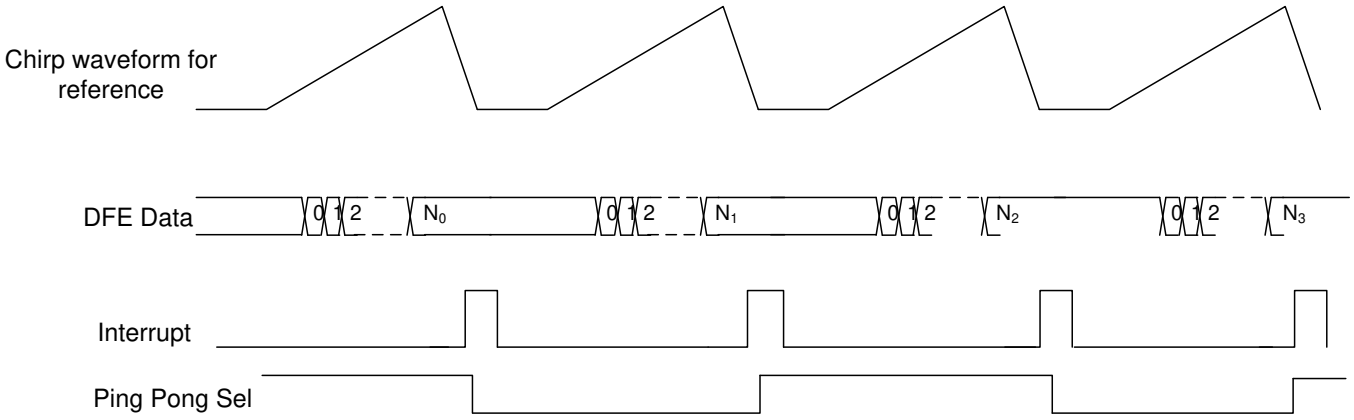
The ADC buffer can be written from DFE in any of the three modes by configuring the control registers ADCBUFCFG1, ADCBUFCFG2, ADCBUFCFG3, and ADCBUFCFG4 in DSS_REG address space:

1. Single-chirp mode
2. Multi-chirp mode
3. Continuous mode

The DFE data from the four Rx channels can be independently enabled by programming the register ADCBUFCFG1.

In single-chirp mode, the FMCW chirp data from the DFE is written to the ADC buffer on a per chirp basis, and a chirp available interrupt is generated on the completion of the write data operation at the end of the chirp, as shown in Figure 13-3. ADC buffer control logic generates the Ping_Pong_Sel signal, as shown in Figure 13-3, which controls whether the data is written into either ping or pong buffer. Data write can start from either the ping or pong buffer.

Figure 13-3. Single-Chirp Mode



In multi-chirp mode, ADC samples for N chirps are stored in a ping/pong buffer before the Ping Pong Select toggles and the Chirp Available Interrupt is generated. The number of chirps stored in the ping buffer is configured in the register field ADCBUFNUMCHRPPING, and the number of chirps to be stored in the pong buffer is configured in the register field ADCBUFNUMCHRPPONG.

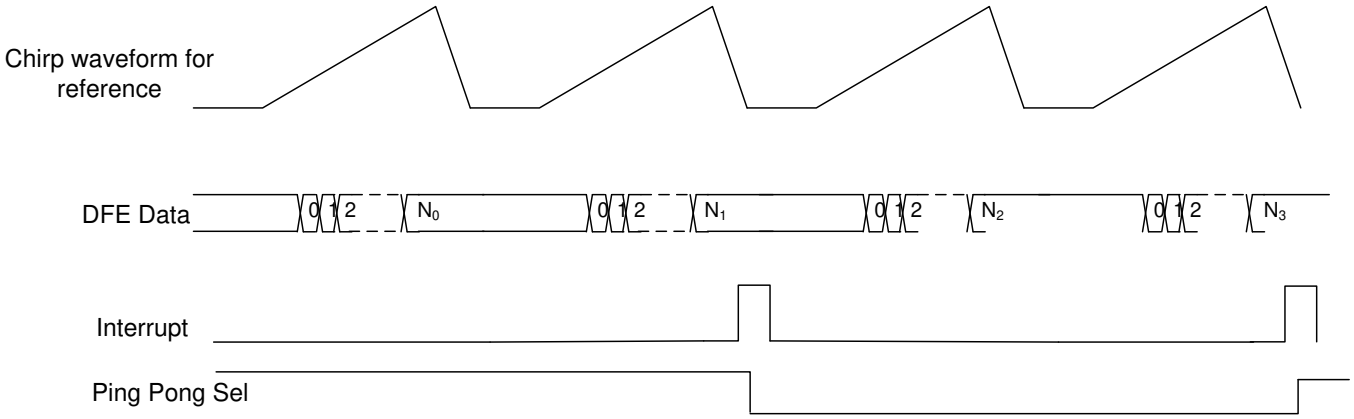
Table 13-1 shows the programming sequence for the ADC buffer single-chirp and multi-chirp modes.

NOTE: Registers for ping and pong must be programmed with the same value for correct functionality.

Table 13-1. ADC Buffer Single-Chirp and Multi-Chirp Mode Programming Sequence

Steps	Register/Bit Field/Programming
Enable the Rx channels for which data will be captured in the ADC buffer	ADCBUFCFG1.RX0EN
	ADCBUFCFG1.RX1EN
	ADCBUFCFG1.RX2EN
	ADCBUFCFG1.RX3EN
Configure the ADC samples to be stored in interleaved or non-interleaved mode	ADCBUFCFG1.ADCBUFWRITEMODE
Configure whether real or complex samples are to be stored in the ADC buffer	ADCBUFCFG1.ADCBUFREALONLYMODE
For complex samples. Configure if I or Q sample is stored in the LSB	ADCBUFCFG1.ADCBUFIQSWAP
Configure the number of samples to be stored in each ping/pong buffer as N	ADCBUFCFG4.ADCBUFNUMCHRPPONG
	ADCBUFCFG4.ADCBUFNUMCHRPP1NG

Figure 13-4. Multi-Chirp Mode



In continuous mode, where the FMCW transceiver is configured to output a single frequency tone in the range of 76-81 GHz, 'N' ADC samples are stored in a ping/pong buffer before the Ping Pong Select toggles and the Chirp Available Interrupt is generated. The value N is configured in the field DSS_REG.ADCBUFCFG4.ADCBUFSAMPCNT. In real mode, this value N refers to the number of real samples per channel, and in complex mode, this refers to the number of complex samples per channel. This counter increments once for every new sample (as long as 1 or more Rx channels are enabled). Continous mode is expected to be only used for CZ and ADC buffer testpattern mode.

Table 13-2 shows the programming sequence for ADC buffer continuous mode.

Table 13-2. ADC Buffer Continuous Mode Programming Sequence

Steps	Register/Bit Field/Programming
Enable the Rx channels for which data will be captured in the ADC buffer	ADCBUFCFG1.RX0EN
	ADCBUFCFG1.RX1EN
	ADCBUFCFG1.RX2EN
	ADCBUFCFG1.RX3EN
Configure the ADC samples to be stored in interleaved or non-interleaved mode	ADCBUFCFG1.ADCBUFWRITEMODE
Configure whether real or complex samples are to be stored in the ADC buffer	ADCBUFCFG1.ADCBUFREALONLYMODE
For complex samples. Configure if I or Q sample is stored in the LSB	ADCBUFCFG1.ADCBUFIQSWAP
Configure the number of samples to be store in each ping/pong buffer	ADCBUFCFG4.ADCBUFSAMPCNT
Enable the ADC buffer in continuous mode	ADCBUFCFG1.ADCBUFCONTMODEEN
To start the capture of samples in the ADC buffer	ADCBUFCFG1.ADCBUFCONTSTRTPL
To stop the capture of samples in the ADC buffer	ADCBUFCFG1.ADCBUFCONTSTOPPL

13.1.2 Support for Hardware in Loop (HIL)

In hardware in loop mode (supported only in the 16xx/18xx/68xx), the ADC data write from DFE can be bypassed by writing the data from the DMM module. Enable this path by writing to the field DSS_REG.DMMSWINT1.DMMADCBUFWREN. The DMM can now access the DSS_ADCBUF memory through the DSS_ADCBUF_WRITE address space. To toggle the ping/pong buffer, toggle DSS_REG.DMMSWINT1.DMMADCBUFPINPONSEL.

13.1.3 Test Pattern Generator Support

An internal test pattern generator which outputs a ramp pattern helps during the initial software development and debug. The output of this module is muxed with the DFE data before sending it to the ADC buffers, as shown in Figure 13-2. Because this is meant for testing the path from the ADC buffer until the final output through LVDS, the ADC buffer configurations must be set to continuous streaming mode, in which the ping-pong switch is based on the number of samples. The test pattern generator can be configured by programming the register TESTPATTERNVLDCFG in the DSS_REG address space. Additional configurable registers are provided for configuring the ramp pattern output from the test pattern generator, such as offset at the start of ramp, step size, and so forth. Refer to the DSS_REG address space and test pattern generator-related registers for further information.

13.1.4 ADC Buffer Data Formats

The data is written in the following formats to the ADC buffer:

- Interleaved data format (supported only in the 14xx)
- Non-interleaved data format (supported in 14xx and 16xx/18xx/68xx)

13.1.4.1 Interleaved Data Format

In interleaved format of data storage, Sample 0 of all the enabled channels, followed by Sample1 of all the enabled channels, and so forth, are written to the buffer as illustrated below for real/complex data, with different configurations of RX channels enabled.

For 4-channels-Rx channels 0,1, 2, and 3 enabled, complex:

RX3_Q(0)	RX3_I(0)	RX2_Q(0)	RX2_I(0)	RX1_Q(0)	RX1_I(0)	RX0_Q(0)	RX0_I(0)

For 4-channels Rx channels 0,1, 2, and 3 enabled, Real Only:

RX3_I(1)	RX2_I(1)	RX1_I(1)	RX0_I(1)	RX3_I(0)	RX2_I(0)	RX1_I(0)	RX0_I(0)

For 3-channels Rx channels 0,1, and 2 enabled, Complex:

NA	NA	RX2_Q(0)	RX2_I(0)	RX1_Q(0)	RX1_I(0)	RX0_Q(0)	RX0_I(0)

For 3-channels Rx channels 0,1, and 3 enabled, Complex:

NA	NA	RX3_Q(0)	RX3_I(0)	RX1_Q(0)	RX1_I(0)	RX0_Q(0)	RX0_I(0)

For 3-channels Rx channels 0, 2, and 3 enabled, Complex:

NA	NA	RX3_Q(0)	RX3_I(0)	RX2_Q(0)	RX2_I(0)	RX0_Q(0)	RX0_I(0)

For 3-channels Rx channels 1, 2, and 3 enabled, Complex:

NA	NA	RX3_Q(0)	RX3_I(0)	RX2_Q(0)	RX2_I(0)	RX1_Q(0)	RX1_I(0)

For 3-channels-Rx channels 0, 1, and 2 enabled, Real only:

NA	NA	RX2_I(1)	RX1_I(1)	RX0_I(1)	RX2_I(0)	RX1_I(0)	RX0_I(0)

For 3-channels Rx channels 0, 1, and 3 enabled, Real only:

NA	NA	RX3_I(1)	RX1_I(1)	RX0_I(1)	RX3_I(0)	RX1_I(0)	RX0_I(0)

For 3-channels Rx channels 0, 2, and 3 enabled, Real only:

NA	NA	RX3_I(1)	RX2_I(1)	RX0_I(1)	RX3_I(0)	RX2_I(0)	RX0_I(0)

For 3-channels Rx channels 1, 2, and 3 enabled, Real only:

NA	NA	RX3_I(1)	RX2_I(1)	RX1_I(1)	RX3_I(0)	RX2_I(0)	RX1_I(0)

For 2-channels Rx channels 0, 1 enabled, Complex:

RX1_Q(1)	RX1_I(1)	RX0_Q(1)	RX0_I(1)	RX1_Q(0)	RX1_I(0)	RX0_Q(0)	RX0_I(0)

For 2-channels Rx channels 0, 1 enabled, Real only:

RX1_I(3)	RX0_I(3)	RX1_I(2)	RX0_I(2)	RX1_I(1)	RX0_I(1)	RX1_I(0)	RX0_I(0)

For 1-channels Rx channels 0 enabled, Complex:

RX0_Q(3)	RX0_I(3)	RX0_Q(2)	RX0_I(2)	RX0_Q(1)	RX0_I(1)	RX0_Q(0)	RX0_I(0)

For 1-channels Rx channels 0 enabled, Real:

RX0_I(7)	RX0_I(6)	RX0_I(5)	RX0_I(4)	RX0_I(3)	RX0_I(2)	RX0_I(1)	RX0_I(0)

Where RXn_I/Q(m) stands for the ‘m’th sample I/Q of ‘n’th RX channel.

13.1.4.2 Non-Interleaved Data Format

In non-interleaved mode storage, each channel data is stored in different memory locations, as shown in Table 13-3.

Table 13-3. Non-Interleaved Data Format

RX0(3)	RX0(2)	RX0(1)	RX0(0)
RX0(7)	RX0(6)	RX0(5)	RX0(4)
RX1(3)	RX1(2)	RX1(1)	RX1(0)
RX1(7)	RX1(6)	RX1(5)	RX1(4)
RX2(3)	RX2(2)	RX2(1)	RX2(0)
RX2(7)	RX2(6)	RX2(5)	RX2(4)
RX3(3)	RX3(2)	RX3(1)	RX3(0)
RX3(7)	RX3(6)	RX3(5)	RX3(4)

High-Speed Interface (HSI)

Topic **Page**

14.1 Overview **1852**
14.2 CBUFF **1852**
14.3 LVDS **1859**
14.4 CSI2 ..... **1998**

14.1 Overview

High-speed interfaces available on the device are mentioned in the HSI integration chapter specific to the device. System data, such as the ADC buffer raw data, chirp profile data, chirp quality data, safety monitoring data, and software variables may be sent out over the high-speed interface.

All data from the system which must be transmitted over CSI2 or LVDs is first sent to the CBUFF controller (Common Buffer Controller). In its simplest form, the CBUFF is an intermediate buffer between the system and the high-speed interfaces, and sits as a slave port on the DSS interconnect.

The CBUFF controller can directly trigger the EDMAs on selected hardware events to request data and initiate an HSI transmission without processor intervention. Alternatively, the CBUFF can also be triggered from software. An EDMA is always used to send data to the CBUFF. This ensures proper synchronization and throughput. The EDMA reads data from any of the data sources and writes to the CBUFF slave port. The CBUFF uses bus flow control mechanisms to handle the differences between EDMA writes and high-speed bandwidth.

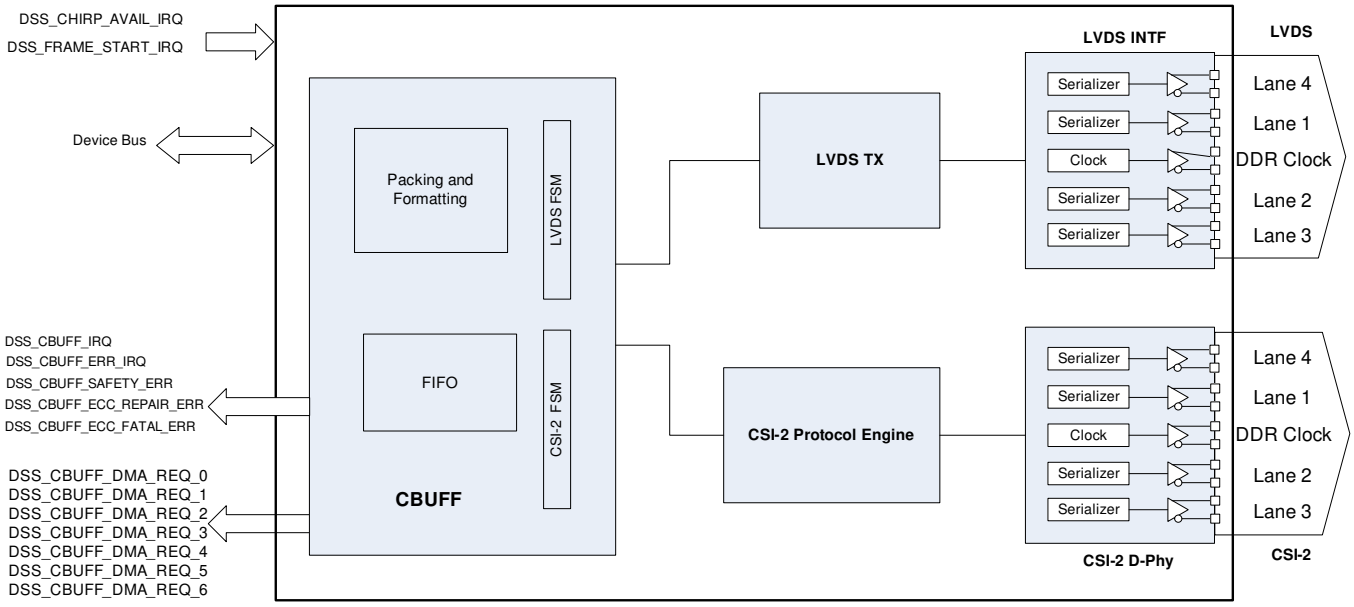
The data from the various sources, such as ADC buffer, CQ memory, and so forth may need to be packed into the correct format before being sent over LVDS and CSI2. Data from multiple sources may also be needed to be combined and sent as part of a single LVDS frame or CSI2 packet. The data may need to be sent between markers such as CSI2 sync packets. All this is done within the CBUFF.

For CSI2, the CBUFF sends the data to the CSI2 protocol engine, which is configured in command mode – OCP mode. The CSI2 protocol engine then sends the data to the CSI2 D-Phy transmitter.

For LVDS, the LVDS TX IP is closely integrated with the configuration of CBUFF, and does not need to be configured separately.

Figure 14-1 illustrates the integration scheme, which shows all the components for HSI data transfer.

Figure 14-1. HSI Overview



NOTE: High-speed interfaces available on the device are mentioned in the HSI integration chapter specific to the device.

14.2 CBUFF

14.2.1 CBUFF Overview

- 128-bit Slave write-only interface over the EDMA writes the data from different sources, such as ADC buffer, chip quality, chirp parameters, safety data, and so forth, to be sent over the high-speed

interfaces. Minimum packet size is 64 bytes.

- Data can be transmitted over any one of the following high-speed interfaces:
  - CSI-2
  - LVDS

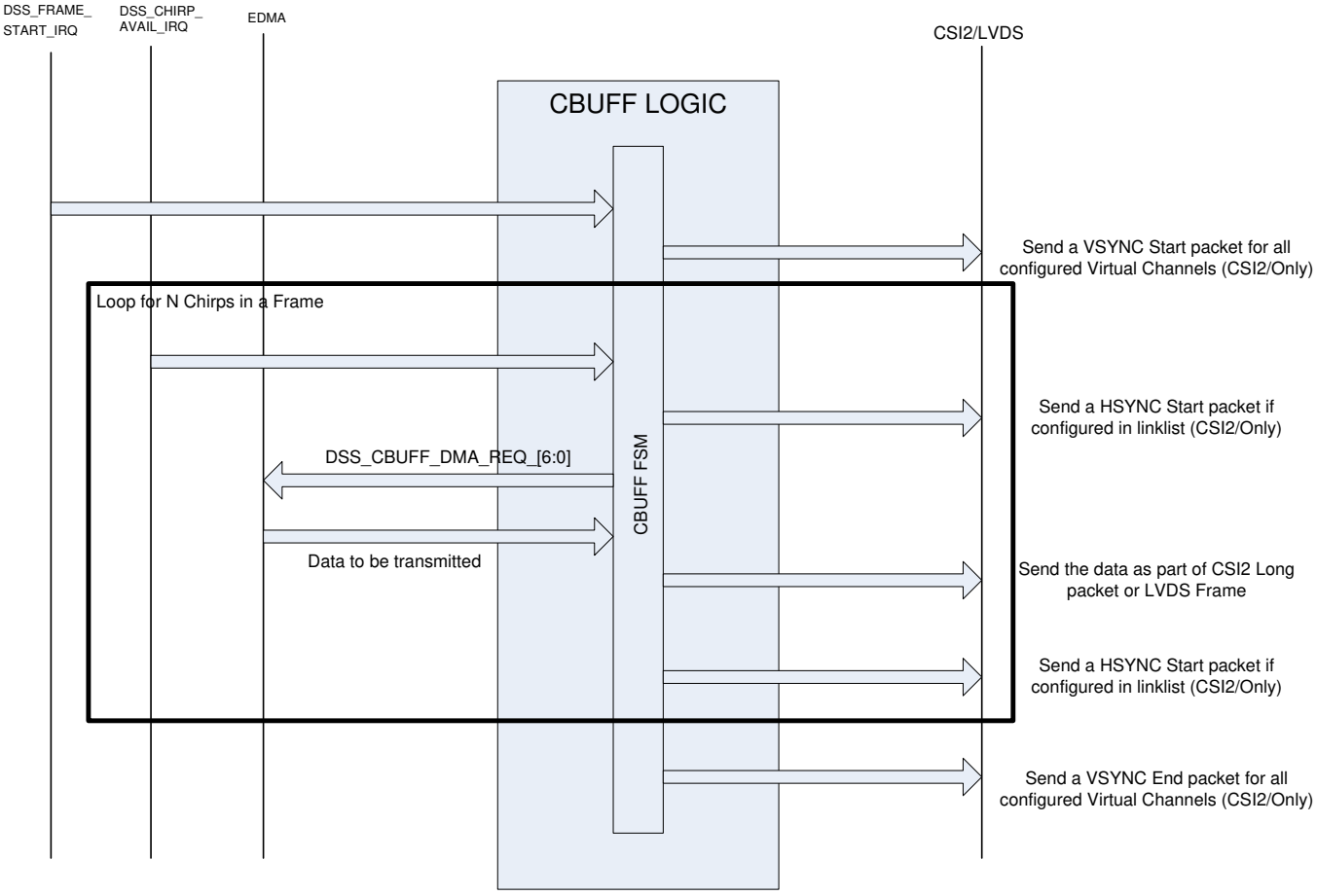
    The CBUFF supports sending the data over up to 4 lanes.
- The data written by the EDMA is stored in a 128-bit internal FIFO. The CBUFF has flow control mechanisms to stall the DMA writes until there is sufficient space in the FIFO to manage bandwidth differences in EDMA and the high-speed interface.
- CBUFF always handles external data at a 16-bit boundary. This is a CBUFF unit: 1 CBUFF UNIT == 16 bits of data
- Minimum size of the LVDS or CSI2 packet is 64 bytes.
- CBUFF supports the following formats:
  - DATA16 : Entire CBUFF unit is sent over CSI/LVDS
  - DATA14 : 14 bits of 1 CBUFF unit is sent over CSI/LVDS
  - DATA12 : 12 bits of 1 CBUFF unit is sent over CSI/LVDS
- Supports packing data from different sources and transmitting as a single LVDS/CSI2 packet through link-list configuration
- CSI2 features:
  - Packing of data into RAW12/RAW14/RAW8 to send data formats DATA12/DATA14/DATA16, respectively
  - Supports virtual channels
  - Supports sending Vsync and Hsync packets
- LVDS features:
  - SDR and DDR clocking modes
  - CRC with the LVDS frame
- Safety features:
  - CRC check on the data transfer between the ADC buffer and CBUFF
  - ECC with single-bit correction and double-bit detection on the CBUFF FIFO
  - Detection of erroneous incomplete transfer when the trigger for the next transfer is received

14.2.2 CBUFF Sequencing

The CBUFF can transmit data over either CSI2 or LVDS. This is selected by configuring the field CONFIG_REG_0.CFG_1LVDS_0CSI.

The CBUFF has an internal state machine that can sequence the transmission of high-speed data without processor intervention. The hardware system events Frame Start (DSS_FRAME_START_IRQ) and Chirp Available (DSS_CHIRP_AVAIL_IRQ) can be used to trigger the data transmission.

Figure 14-2. CBUFF Sequencing



14.2.2.1 LVDS Transmission Sequence

1. CBUFF is programmed and enabled.
2. On Frame Start, CBUFF moves to the next state to wait for the Chirp Available trigger.
3. On Chirp Available, CBUFF generates an EDMA hardware request. The EDMA sends the data to be sent over HSI to CBUFF, and CBUFF transfers the data over LVDS. On completion, it sends a Chirp Done interrupt.
4. If this is the last Chirp configured in the CBUFF, it generates a Frame Done interrupt and waits for the next Frame Sync (Step 2). Else, it waits for the next Chirp available (Step 3).

14.2.2.2 CSI2 Transmission Sequencing

1. CBUFF is programmed and enabled.
2. On Frame Start, CBUFF sends a Vertical Start Sync Short packet command to the CSI protocol engine, and waits for a Packet Sent signal from the CSI protocol engine.
3. On Chirp Available, it generates a DMA request signal and sends a Horizontal Start Short Packet command to the CSI protocol engine, which is configured in Command Mode-OCP, then waits for the Packet Sent signal from the CSI protocol engine. The DMA begins to send the data to be transferred over HSI to the CBUFF.
4. CBUFF sends a Long Packet Header command to the CSI protocol engine.
5. CBUFF writes the data to be transferred, filled in its FIFO to the Long Packet Payload register.
6. CBUFF waits for Packet Sent, then sends the Horizontal End Sync Packet command to the CSI PE, and waits for the Packet Sent signal from the CSI PE before sending a Chirp Done interrupt.

7. If this is the last Chirp configured in the CBUFF, it sends a Vertical End Sync Short packet command to CSI PE, waits for the Packet Sent signal from the CSI PE, sends a Frame Done Interrupt, then waits for the next Frame Sync (Step 2). Else, it waits for the next Chirp available (Step 3).

14.2.3 Sequence-Related CBUFF Configuration Fields

14.2.3.1 Software Triggering Option

CBUFF has the option to be triggered by software, by configuring the fields CONFIG_REG_0.CFG_SW_TRIG_EN and CONFIG_REG_0.CFTRIGEN. If these fields are configured for the software trigger, then the CBUFF can be triggered by writing into the field CONFIG_REG_0.CFG_FRAME_START_TRIG to generate a Frame Start trigger and CONFIG_REG_0. CFG_CHIRP_AVAIL_TRIG to generate a Chirp Available trigger to the CBUFF.

14.2.3.2 Configuration for CSI2 VSYNC Packet Transmission (CSI2 Only)

CBUFF can be configured to send out VSYNC Start and End packets on any of the four CSI2 virtual channel IDs. Configure CONFIG_REG_0.CVC0EN for VSYNC packets on Virtual Channel 0; configure CONFIG_REG_0.CVC1EN for VSYNC packets on Virtual Channel 1, and so forth.

14.2.3.3 Number of Chirps per Frame

The register CFG_CHIRPS_PER_FRAME determines the number of times the CBUFF state machine loops as shown in Figure 14-2.

14.2.3.4 Configuring the Short Packet Details

The CSI2 sync packets codes and the CSI2 Protocol Engine Register mapping are driven from registers, and must be configured before transmission. Though there are no sync packets in LVDS, these registers (CFG_SPHDR_ADDRESS, CFG_CMD_HSVAL, CFG_CMD_HEVAL, CFG_CMD_VSVAL, CFG_CMD_VEVAL, CFG_LPHDR_ADDRESS, and CFG_LPPYLD_ADDRESS) must be configured appropriately as per details provided in their descriptions.

14.2.4 CBUFF Linklist Concept

Figure 14-3. CBUFF Linklist Concept

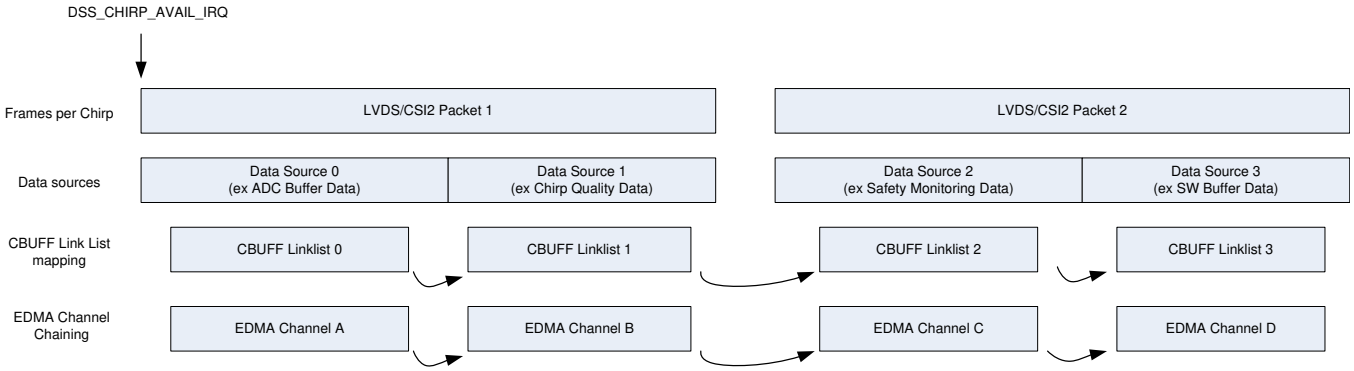


Figure 14-3 shows how data to be sent out within a Chirp is broken down in the system and spilt into configurations for the different IPs involved in the data transmission.

Each of the data sources has a separate CBUFF linklist entry in the CBUFF.

The following sections list the configurable parameters available for each linklist. For more details, refer to the *AWR1xxx Data Path Programmer's Guide* API document.

14.2.4.1 VALID

The field CFG_DATA_LL[X].LL[X]_VALID determines if the linklist entry is valid. Once the CBUFF is triggered for data transmission, the CBUFF linklist is parsed from Linklist 0 to Linklist 31 until an invalid Linklist entry is found. When an invalid linklist is found, there should not be any further valid entries.

14.2.4.2 HSYNC Start and HSYNC End

For CSI2 transmission, the field CFG_DATA_LL[X]. LL[X]_HS determines if a HSYNC start packet is sent out before the transmission of Linklist data. This field can be set to 0x1 only of the linklist is the start of a new CSI2 packet.

For CSI2 transmission, the field CFG_DATA_LL[X]. LL[X]_HE determines if a HSYNC end packet is sent out after the transmission of data for CSI2. This field can be set to 0x1 only of the linklist is the end of a CSI2 packet.

14.2.4.3 Virtual Channel Number

For CSI2 transmission, the field CFG_DATA_LL[X]. LL[X]_VCNUM is used as the virtual channel number for the HSYNC packet related to the linklist.

14.2.4.4 CRC Enable

The field CFG_DATA_LL[X].LL[X]_CRC_EN is related to the safety feature on data from the ADC buffer. CBUFF can perform a CRC check on the data being sent from the ADC buffer. If the ADC buffer is stored in interleaved format, the entire ADC buffer data should be mapped to a single CBUFF linklist entry. If the ADC buffer is stored in non-interleaved format, each Rx channel should be mapped to a separate CBUFF linklist entry with the field set in each of the linklists. Also, Rx[n] channel data sent out before Rx[m] channel data. Where n < m.

14.2.4.5 Format

CBUFF always handles external data at 16-bit boundary. This is a CBUFF unit:

1 CBUFF UNIT == 16 bits of data

CBUFF supports the following formats:

- DATA16 : Entire CBUFF unit is sent over CSI/LVDS
- DATA14 : 14 bits of 1 CBUFF unit is sent over CSI/LVDS
- DATA12 : 12 bits of 1 CBUFF unit is sent over CSI/LVDS

If fewer than 16 bits of data are sent out, the LSB bits of the CBUFF unit are sent out. The format is determined by the field CFG_DATA_LL[X].LL[X]_FMT. The format for all linklist within a CSI packet/ LVDS frame remains the same.

14.2.4.6 LVDS Format Mapping

The linklist-independent registers CFG_LVDS_MAPPING_LANEx_FMT_0 and CFG_LVDS_MAPPING_LANEx_FMT_1 are used to configure the mapping of each CBUFF unit (16-bit data) within the 128-bit CBUFF FIFO line to a specific LVDS lane. The field LL[X]_FMT_MAP in each CFG_DATA_LL[X] register selects whether FMT0 or FMT1 should be used for that linklist entry. All linklist entries must use the same format (0 or 1).

Consider the configuration for CFG_LVDS_MAPPING_LANEx_FMT_0. There are 4 such registers for each of the LVDS lanes:

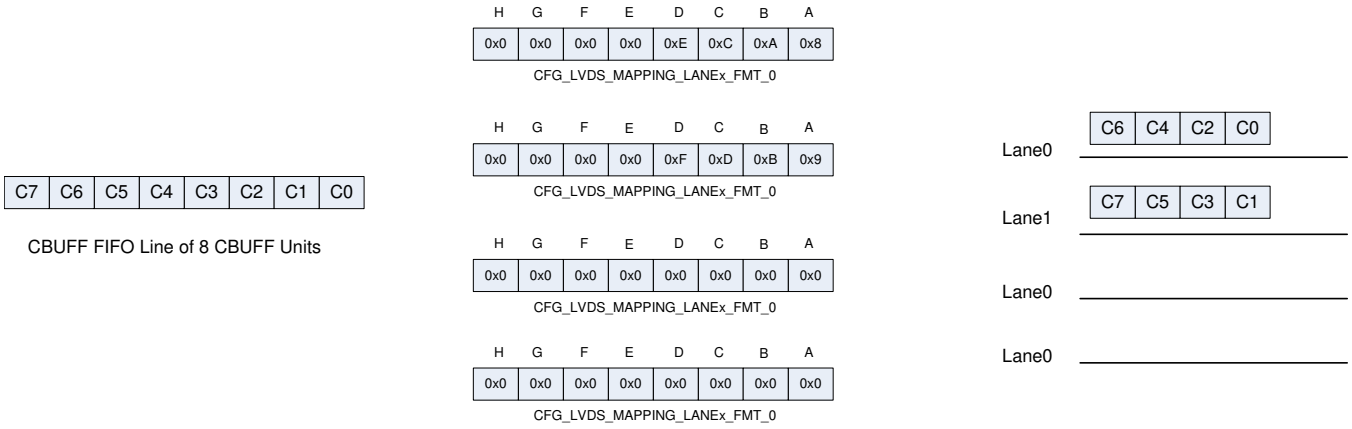
- CFG_LVDS_MAPPING_LANE0_FMT_0
- CFG_LVDS_MAPPING_LANE1_FMT_0
- CFG_LVDS_MAPPING_LANE2_FMT_0
- CFG_LVDS_MAPPING_LANE3_FMT_0

Consider the register for a LVDS Lane 0, CFG_LVDS_MAPPING_LANE0_FMT_0. It has 8 fields, CFG_LVDS_MAPPING_LANE0_FMT_1_A to CFG_LVDS_MAPPING_LANE3_FMT_1_H.

For each of the 4-bit fields, Bits [2:0] selects 1 of the 8 CBUFF units in a CBUFF FIFO line, which is mapped to this lane. Bit[3] is set to 0x1 to indicate it is a valid mapping.

Figure 14-4 is an example of the default use case for the 16xx/18xx/68xx device, where data is sent over 2 LVDS lanes. For more details, refer to the *AWR1xxx Data Path Programmer's Guide* API document.

Figure 14-4. Default Use Case



Where x is 0-3.

14.2.4.7 Input Data Format

The field CFG_DATA_LL[X].LL[X]_FMT_IN allows user to configure if all 128 bits in a CBUFF FIFO line are sent out over the interface, or if only the lower 96 bits in a 128-bit CBUFF FIFO line are send out over HSI and the rest are discarded. This is useful for 3-channel data stored in interleaved mode. For more details, refer to the *AWR1xxx Data Path Programmer's Guide* API document.

14.2.4.8 Size

The field CFG_DATA_LL[X].LL[X]_SIZE determines the size of the data for the given linklist. The value configured is the number of CBUFF units of data for the given linklist.

The minimum size of a link list is 2 CBUFF units. The maximum size is [16K -1] CBUFF units.

14.2.4.9 Long Packet Header

If the linklist is the start of a CSI2 packet, the field CFG_DATA_LL[X]_LPHDR_EN should be set to 0x1. When this field is set, the data in the register CFG_DATA_LL[X]_LPHDR_VAL is sent as the long packet header for the CSI2 packet. The long packet 32-bit header should be calculated as in Equation 2.

0xEC << 24 | [CSI2 packet size in bytes] << 8) | CSI2 Data Packet ID (2)

For more details, refer to the *AWR1xxx Data Path Programmer's Guide* API document.

14.2.4.10 EDMA Request Number

For Linklist 0, then CBUFF generates a DMA request to trigger the DMA transfer to trigger the transfer of data to be transmitted out for the chirp. The DMA request number to the EDMA can be configured in CFG_DATA_LL0_THRESHOLD.LL0DMAN.

14.2.5 CBUFF Interrupts

14.2.5.1 CBUFF Interrupt (DSS_CBUFF_IRQ)

The CBUFF interrupt can be generated on completion of data transmission for a chirp. To enable this interrupt generation from CBUFF, the field CFG_MASK_REG0[11] must be unmasked by clearing it. On an interrupt, the processor must read the STAT_CBUFF_REG0.S_CHIRP_DONE interrupt to confirm the event occurred. The processor should then clear the interrupt by writing 0x1 into the CLR_CBUFF_REG0.C_CHIRP_DONE field.

The CBUFF interrupt can be generated on completion of data transmission for a frame. To enable this interrupt generation from CBUFF, the field CFG_MASK_REG0[12] must be unmasked by clearing it. On an interrupt, the processor must read the STAT_CBUFF_REG0.S_FRAME_DONE interrupt to confirm the event occurred. The processor should then clear the interrupt by writing 0x1 into the CLR_CBUFF_REG0.C_FRAME_DONE field.

14.2.5.2 CBUFF Error Interrupt (DSS_CBUFF_ERR_IRQ)

The CBUFF Error interrupt can be generated if the CBUFF receives another Chirp Available hardware event trigger before completion of data transmission for the previous chirp. To enable this interrupt generation from CBUFF, the field CFG_MASK_REG1[16] must be unmasked by clearing it. On an interrupt, the processor must read the STAT_CBUFF_REG1.S_CHIRP_ERR interrupt to confirm the event occurred. The processor should then clear the interrupt by writing 0x1 into the CLR_CBUFF_REG1[16] field.

The CBUFF Error interrupt can be generated if the CBUFF receives another Frame Start hardware event trigger before completion of data transmission for the previous frame. To enable this interrupt generation from CBUFF, the field CFG_MASK_REG1[17] must be unmasked by clearing it. On an interrupt, the processor must read the STAT_CBUFF_REG1.S_FRAME_ERR interrupt to confirm the event occurred. The processor should then clear the interrupt by writing 0x1 into the CLR_CBUFF_REG1[17] field.

14.2.5.3 CBUFF Safety Interrupt (DSS_CBUFF_SAFETY_ERR)

The CBUFF Safety interrupt can be generated if the CBUFF receives another Chirp Available hardware event trigger before completion of data transmission for the previous chirp. To enable this interrupt generation from CBUFF, the field MASK_SAFETY[8] must be unmasked by clearing it. On an interrupt, the processor must read the STAT_SAFETY.SAF_CHIRP_ERR interrupt to confirm the event occurred. The processor should then clear the interrupt by writing 0x1 into the CLR_SAFETY[8] field.

The CBUFF Safety interrupt can be generated if the CBUFF detects a CRC error on the data from ADC buffer. To enable this interrupt generation from CBUFF, the field MASK_SAFETY[7:0] must be unmasked by clearing it. On an interrupt, the processor must read the STAT_SAFETY.SAF_CRC interrupt to confirm the event occurred. The processor should then clear the interrupt by writing 0x1 into the CLR_SAFETY[7:0] bits.

14.2.5.4 CBUFF ECC Interrupts (DSS_CBUFF_ECC_REPAIR_ERR and DSS_CBUFF_ECC_FATAL_ERR)

CBUFF can be enabled with ECC on its internal 128-bit wide FIFO by configuring the CONFIG_REG_0.CFG_ECC_EN field. To enable the interrupts to ESM for Single Bit Error (SBE) correction and Double Bit Error (DBE) detection, the interrupts should be unmasked by clearing the fields MASK_CBUFF_ECC_REG.MECCSBE and MASK_CBUFF_ECC_REG.MECCDBE. On an interrupt, processor must read the fields STAT_CBUFF_ECC_REG.SECCSBE and STAT_CBUFF_ECC_REG.SECCDBE to confirm an error was generated. If a status bit is set, then the field STAT_CBUFF_ECC_REG.SECCADD provides the FIFO address where the error occurred. The processor must then clear the status by writing to the register CLR_CBUFF_ECC_REG.CECCSBE and CLR_CBUFF_ECC_REG.CECCDBE and CLR_CBUFF_ECC_REG.CECCADD fields.

CBUFF programming: see the relevant CBUFF register programming in CSI2 and LVDS programming sections.

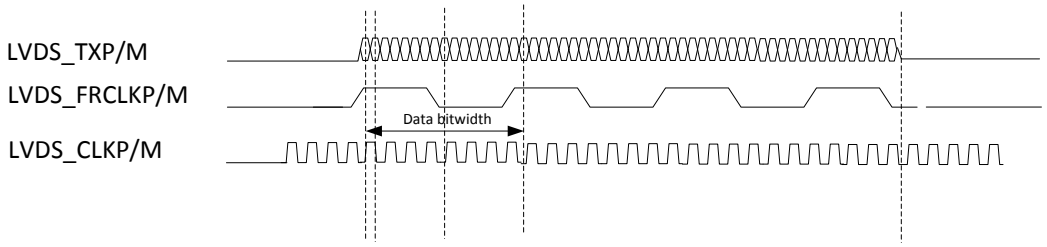
14.3 LVDS

14.3.1 LVDS Overview

The LVDS interface includes the following signals:

- LVDS bit clock
- LVDS data lanes (the HSI integration chapter specific to the device lists the number of available lanes specific to the device)
- LVDS frame clock
- LVDS data_valid signal (not supported in the 16xx/18xx/68xx)

Figure 14-5. LVDS Interface Timings



The LVDS interface supports the following data rates:

- 900 Mbps (450-MHz DDR Clock)
- 600 Mbps (300-MHz DDR Clock)
- 450 Mbps (225-MHz DDR Clock)
- 400 Mbps (200-MHz DDR Clock)
- 300 Mbps (150-MHz DDR Clock)
- 225 Mbps (112.5-MHz DDR Clock)
- 150 Mbps (75-MHz DDR Clock)

Refer to the device data sheet for more details.

14.3.2 LVDS Programming Sequence

The following sections show the programming sequence needed before the hardware triggers are generated to initiate the high-speed LVDS data transmission.

14.3.2.1 LVDS Global Initialization

Table 14-1. Main Sequence – PRCM and Global Configuration

Steps	Register/Bit Field/Programming	Value
Power on the LVDS I/Os	MSS_TOP_RCM. LVDSPADCTL0 MSS_TOP_RCM. LVDSPADCTL1	0x0 0x0
Power off the LVDS I/Os	MSS_TOP_RCM. LVDSPADCTL0 MSS_TOP_RCM. LVDSPADCTL1	0x01010101 0x010101
Power on the LVDS I/Os	MSS_TOP_RCM. LVDSPADCTL0 MSS_TOP_RCM. LVDSPADCTL1	0x0 0x0

14.3.2.2 CBUFF Configuration

Table 14-2. Main Sequence – CBUFF LVDS Static Configuration

Steps	Register/Bit Field/Programming	Value
Assert the CBUFF soft reset	CONFIG_REG_0.CSWCRST	0x1
Configure CBUFF for LVDS data transfer	CONFIG_REG_0.CFG_1LVDS_0CSI__POS	0x1
Configure static values for LVDS	CONFIG_REG_0.CVC0EN	
	CFG_SPHDR_ADDRESS	0x3
	CFG_CMD_VSVAL	0x55555555
	CFG_CMD_VEVAL	0xAAAAAAAA
	CFG_LPHDR_ADDRESS	0xAAAAAAAA
	CFG_LVDS_GEN_0.CCSMEN	0x1
Configure the number of chirps in a frame	CFG_CHIRPS_PER_FRAME	X
Configure static values for LVDS based on LVDS CRC enabled or disabled	CFG_LVDS_GEN_0.CBCRCEN CFG_CMD_HEVAL CFG_CMD_HSVAL	X
Enable the LVDS lanes	CFG_LVDS_GEN_0.CFG_LVDS_LANE[X]_EN	0x1
Set the SDR or DDR mode	CFG_LVDS_GEN_0.CFG_BIT_CLK_MODE	0x-
Set the SDR or DDR mode clock mux	CFG_LVDS_GEN_0.CCLKSEL1	0x-
Configure the alignment for start of samples	CFG_LVDS_GEN_0.CPOSSEL	0x-
Configure the LVDS FIFO initial threshold	CFG_LVDS_GEN_0.CFDLY	0x8
Set the 3C3L mode if the system configuration is interleaved 3 channel – 3 lane	CFG_LVDS_GEN_1.C3C3L	0x-
Configure the lane-mapping format registers	CFG_LVDS_MAPPING_LANE[X]_FMT_0 CFG_LVDS_MAPPING_LANE[X]_FMT_1	X
Release the CBUFF from soft reset	CONFIG_REG_0.CSWCRST	0x0

The configuration in Table 14-3 should be performed for each linklist entry required to transmit the LVDS packet.

Table 14-3. Main Sequence – CBUFF Linklist

Steps	Register/Bit Field/Programming	Value
Set the valid for the linklist	CFG_DATA_LL[X].LL[X]_VALID	0x1
If the linklist is the start of a new LVDS packet	CFG_DATA_LL[X].LL[X]_LPHDR_EN CFG_DATA_LL[X].LL[X]_HS	0x-
Configure the long packet header to static value for LVDS	CFG_DATA_LL[X]_LPHDR_VAL	0xBBBBBBBB
If the linklist is the end of a LVDS packet	CFG_DATA_LL[X].LL[X]_HE	0x-
Configure the size in CBUFF units	CFG_DATA_LL[X].LL[X]_SIZE	X
Configure the format of the CSI2 packet to which the linklist belongs	CFG_DATA_LL[X].LL[X]_FMT	X
Select the LVDS format-mapping register for the LVDS packet	CFG_DATA_LL[X].LL[X]_FMT_MAP	0x-
Set the input format	CFG_DATA_LL[X].LL[X]_FMT_IN	X
Set the Linklist write threshold	CFG_DATA_LL[X]_THRESHOLD.LL[X]_WR_THRESHOLD	X

Table 14-3. Main Sequence – CBUFF Linklist (continued)

Steps	Register/Bit Field/Programming	Value
Set the Linklist read threshold	CFG_DATA_LL[X]_THRESHOLD.LL[X]_WR_THRESHOLD	X

14.3.3 CBUFF and LVDS Registers

14.3.3.1 DSS_CBUFF Registers

Table 14-4 lists the memory-mapped registers for the DSS_CBUFF. All register offset addresses not listed in Table 14-4 should be considered as reserved locations and the register contents should not be modified.

Table 14-4. DSS_CBUFF Registers

Offset	Acronym	Register Name	Section
0h	CONFIG_REG_0	CONFIG_REG_0	Section 14.3.3.1.1
4h	CFG_SPHDR_ADDRESS	CFG_SPHDR_ADDRESS	Section 14.3.3.1.2
8h	CFG_CMD_HSVAL	CFG_CMD_HSVAL	Section 14.3.3.1.3
Ch	CFG_CMD_HEVAL	CFG_CMD_HEVAL	Section 14.3.3.1.4
10h	CFG_CMD_VSVAL	CFG_CMD_VSVAL	Section 14.3.3.1.5
14h	CFG_CMD_VEVAL	CFG_CMD_VEVAL	Section 14.3.3.1.6
18h	CFG_LPHDR_ADDRESS	CFG_LPHDR_ADDRESS	Section 14.3.3.1.7
20h	CFG_CHIRPS_PER_FRAME	CFG_CHIRPS_PER_FRAME	Section 14.3.3.1.8
24h	CFG_FIFO_FREE_THRESHOLD	CFG_FIFO_FREE_THRESHOLD	Section 14.3.3.1.9
28h	CFG_LPPYLD_ADDRESS	CFG_LPPYLD_ADDRESS	Section 14.3.3.1.10
30h	CFG_DATA_LL0	CFG_DATA_LL0	Section 14.3.3.1.11
34h	CFG_DATA_LL0_LPHDR_VAL	CFG_DATA_LL0_LPHDR_VAL	Section 14.3.3.1.12
38h	CFG_DATA_LL0_THRESHOLD	CFG_DATA_LL0_THRESHOLD	Section 14.3.3.1.13
3Ch	CFG_DATA_LL1	CFG_DATA_LL1	Section 14.3.3.1.14
40h	CFG_DATA_LL1_LPHDR_VAL	CFG_DATA_LL1_LPHDR_VAL	Section 14.3.3.1.15
44h	CFG_DATA_LL1_THRESHOLD	CFG_DATA_LL1_THRESHOLD	Section 14.3.3.1.16
48h	CFG_DATA_LL2	CFG_DATA_LL2	Section 14.3.3.1.17
4Ch	CFG_DATA_LL2_LPHDR_VAL	CFG_DATA_LL2_LPHDR_VAL	Section 14.3.3.1.18
50h	CFG_DATA_LL2_THRESHOLD	CFG_DATA_LL2_THRESHOLD	Section 14.3.3.1.19
54h	CFG_DATA_LL3	CFG_DATA_LL3	Section 14.3.3.1.20
58h	CFG_DATA_LL3_LPHDR_VAL	CFG_DATA_LL3_LPHDR_VAL	Section 14.3.3.1.21
5Ch	CFG_DATA_LL3_THRESHOLD	CFG_DATA_LL3_THRESHOLD	Section 14.3.3.1.22
60h	CFG_DATA_LL4	CFG_DATA_LL4	Section 14.3.3.1.23
64h	CFG_DATA_LL4_LPHDR_VAL	CFG_DATA_LL4_LPHDR_VAL	Section 14.3.3.1.24
68h	CFG_DATA_LL4_THRESHOLD	CFG_DATA_LL4_THRESHOLD	Section 14.3.3.1.25
6Ch	CFG_DATA_LL5	CFG_DATA_LL5	Section 14.3.3.1.26
70h	CFG_DATA_LL5_LPHDR_VAL	CFG_DATA_LL5_LPHDR_VAL	Section 14.3.3.1.27
74h	CFG_DATA_LL5_THRESHOLD	CFG_DATA_LL5_THRESHOLD	Section 14.3.3.1.28
78h	CFG_DATA_LL6	CFG_DATA_LL6	Section 14.3.3.1.29
7Ch	CFG_DATA_LL6_LPHDR_VAL	CFG_DATA_LL6_LPHDR_VAL	Section 14.3.3.1.30
80h	CFG_DATA_LL6_THRESHOLD	CFG_DATA_LL6_THRESHOLD	Section 14.3.3.1.31
84h	CFG_DATA_LL7	CFG_DATA_LL7	Section 14.3.3.1.32
88h	CFG_DATA_LL7_LPHDR_VAL	CFG_DATA_LL7_LPHDR_VAL	Section 14.3.3.1.33
8Ch	CFG_DATA_LL7_THRESHOLD	CFG_DATA_LL7_THRESHOLD	Section 14.3.3.1.34
90h	CFG_DATA_LL8	CFG_DATA_LL8	Section 14.3.3.1.35
94h	CFG_DATA_LL8_LPHDR_VAL	CFG_DATA_LL8_LPHDR_VAL	Section 14.3.3.1.36
98h	CFG_DATA_LL8_THRESHOLD	CFG_DATA_LL8_THRESHOLD	Section 14.3.3.1.37
9Ch	CFG_DATA_LL9	CFG_DATA_LL9	Section 14.3.3.1.38
A0h	CFG_DATA_LL9_LPHDR_VAL	CFG_DATA_LL9_LPHDR_VAL	Section 14.3.3.1.39
A4h	CFG_DATA_LL9_THRESHOLD	CFG_DATA_LL9_THRESHOLD	Section 14.3.3.1.40
A8h	CFG_DATA_LL10	CFG_DATA_LL10	Section 14.3.3.1.41
ACh	CFG_DATA_LL10_LPHDR_VAL	CFG_DATA_LL10_LPHDR_VAL	Section 14.3.3.1.42
B0h	CFG_DATA_LL10_THRESHOLD	CFG_DATA_LL10_THRESHOLD	Section 14.3.3.1.43

Table 14-4. DSS_CBUFF Registers (continued)

Offset	Acronym	Register Name	Section
B4h	CFG_DATA_LL11	CFG_DATA_LL11	Section 14.3.3.1.44
B8h	CFG_DATA_LL11_LPHDR_VAL	CFG_DATA_LL11_LPHDR_VAL	Section 14.3.3.1.45
BCh	CFG_DATA_LL11_THRESHOLD	CFG_DATA_LL11_THRESHOLD	Section 14.3.3.1.46
C0h	CFG_DATA_LL12	CFG_DATA_LL12	Section 14.3.3.1.47
C4h	CFG_DATA_LL12_LPHDR_VAL	CFG_DATA_LL12_LPHDR_VAL	Section 14.3.3.1.48
C8h	CFG_DATA_LL12_THRESHOLD	CFG_DATA_LL12_THRESHOLD	Section 14.3.3.1.49
CCh	CFG_DATA_LL13	CFG_DATA_LL13	Section 14.3.3.1.50
D0h	CFG_DATA_LL13_LPHDR_VAL	CFG_DATA_LL13_LPHDR_VAL	Section 14.3.3.1.51
D4h	CFG_DATA_LL13_THRESHOLD	CFG_DATA_LL13_THRESHOLD	Section 14.3.3.1.52
D8h	CFG_DATA_LL14	CFG_DATA_LL14	Section 14.3.3.1.53
DCh	CFG_DATA_LL14_LPHDR_VAL	CFG_DATA_LL14_LPHDR_VAL	Section 14.3.3.1.54
E0h	CFG_DATA_LL14_THRESHOLD	CFG_DATA_LL14_THRESHOLD	Section 14.3.3.1.55
E4h	CFG_DATA_LL15	CFG_DATA_LL15	Section 14.3.3.1.56
E8h	CFG_DATA_LL15_LPHDR_VAL	CFG_DATA_LL15_LPHDR_VAL	Section 14.3.3.1.57
ECh	CFG_DATA_LL15_THRESHOLD	CFG_DATA_LL15_THRESHOLD	Section 14.3.3.1.58
F0h	CFG_DATA_LL16	CFG_DATA_LL16	Section 14.3.3.1.59
F4h	CFG_DATA_LL16_LPHDR_VAL	CFG_DATA_LL16_LPHDR_VAL	Section 14.3.3.1.60
F8h	CFG_DATA_LL16_THRESHOLD	CFG_DATA_LL16_THRESHOLD	Section 14.3.3.1.61
FCh	CFG_DATA_LL17	CFG_DATA_LL17	Section 14.3.3.1.62
100h	CFG_DATA_LL17_LPHDR_VAL	CFG_DATA_LL17_LPHDR_VAL	Section 14.3.3.1.63
104h	CFG_DATA_LL17_THRESHOLD	CFG_DATA_LL17_THRESHOLD	Section 14.3.3.1.64
108h	CFG_DATA_LL18	CFG_DATA_LL18	Section 14.3.3.1.65
10Ch	CFG_DATA_LL18_LPHDR_VAL	CFG_DATA_LL18_LPHDR_VAL	Section 14.3.3.1.66
110h	CFG_DATA_LL18_THRESHOLD	CFG_DATA_LL18_THRESHOLD	Section 14.3.3.1.67
114h	CFG_DATA_LL19	CFG_DATA_LL19	Section 14.3.3.1.68
118h	CFG_DATA_LL19_LPHDR_VAL	CFG_DATA_LL19_LPHDR_VAL	Section 14.3.3.1.69
11Ch	CFG_DATA_LL19_THRESHOLD	CFG_DATA_LL19_THRESHOLD	Section 14.3.3.1.70
120h	CFG_DATA_LL20	CFG_DATA_LL20	Section 14.3.3.1.71
124h	CFG_DATA_LL20_LPHDR_VAL	CFG_DATA_LL20_LPHDR_VAL	Section 14.3.3.1.72
128h	CFG_DATA_LL20_THRESHOLD	CFG_DATA_LL20_THRESHOLD	Section 14.3.3.1.73
12Ch	CFG_DATA_LL21	CFG_DATA_LL21	Section 14.3.3.1.74
130h	CFG_DATA_LL21_LPHDR_VAL	CFG_DATA_LL21_LPHDR_VAL	Section 14.3.3.1.75
134h	CFG_DATA_LL21_THRESHOLD	CFG_DATA_LL21_THRESHOLD	Section 14.3.3.1.76
138h	CFG_DATA_LL22	CFG_DATA_LL22	Section 14.3.3.1.77
13Ch	CFG_DATA_LL22_LPHDR_VAL	CFG_DATA_LL22_LPHDR_VAL	Section 14.3.3.1.78
140h	CFG_DATA_LL22_THRESHOLD	CFG_DATA_LL22_THRESHOLD	Section 14.3.3.1.79
144h	CFG_DATA_LL23	CFG_DATA_LL23	Section 14.3.3.1.80
148h	CFG_DATA_LL23_LPHDR_VAL	CFG_DATA_LL23_LPHDR_VAL	Section 14.3.3.1.81
14Ch	CFG_DATA_LL23_THRESHOLD	CFG_DATA_LL23_THRESHOLD	Section 14.3.3.1.82
150h	CFG_DATA_LL24	CFG_DATA_LL24	Section 14.3.3.1.83
154h	CFG_DATA_LL24_LPHDR_VAL	CFG_DATA_LL24_LPHDR_VAL	Section 14.3.3.1.84
158h	CFG_DATA_LL24_THRESHOLD	CFG_DATA_LL24_THRESHOLD	Section 14.3.3.1.85
15Ch	CFG_DATA_LL25	CFG_DATA_LL25	Section 14.3.3.1.86
160h	CFG_DATA_LL25_LPHDR_VAL	CFG_DATA_LL25_LPHDR_VAL	Section 14.3.3.1.87
164h	CFG_DATA_LL25_THRESHOLD	CFG_DATA_LL25_THRESHOLD	Section 14.3.3.1.88
168h	CFG_DATA_LL26	CFG_DATA_LL26	Section 14.3.3.1.89
16Ch	CFG_DATA_LL26_LPHDR_VAL	CFG_DATA_LL26_LPHDR_VAL	Section 14.3.3.1.90

Table 14-4. DSS_CBUFF Registers (continued)

Offset	Acronym	Register Name	Section
170h	CFG_DATA_LL26_THRESHOLD	CFG_DATA_LL26_THRESHOLD	Section 14.3.3.1.91
174h	CFG_DATA_LL27	CFG_DATA_LL27	Section 14.3.3.1.92
178h	CFG_DATA_LL27_LPHDR_VAL	CFG_DATA_LL27_LPHDR_VAL	Section 14.3.3.1.93
17Ch	CFG_DATA_LL27_THRESHOLD	CFG_DATA_LL27_THRESHOLD	Section 14.3.3.1.94
180h	CFG_DATA_LL28	CFG_DATA_LL28	Section 14.3.3.1.95
184h	CFG_DATA_LL28_LPHDR_VAL	CFG_DATA_LL28_LPHDR_VAL	Section 14.3.3.1.96
188h	CFG_DATA_LL28_THRESHOLD	CFG_DATA_LL28_THRESHOLD	Section 14.3.3.1.97
18Ch	CFG_DATA_LL29	CFG_DATA_LL29	Section 14.3.3.1.98
190h	CFG_DATA_LL29_LPHDR_VAL	CFG_DATA_LL29_LPHDR_VAL	Section 14.3.3.1.99
194h	CFG_DATA_LL29_THRESHOLD	CFG_DATA_LL29_THRESHOLD	Section 14.3.3.1.100
198h	CFG_DATA_LL30	CFG_DATA_LL30	Section 14.3.3.1.101
19Ch	CFG_DATA_LL30_LPHDR_VAL	CFG_DATA_LL30_LPHDR_VAL	Section 14.3.3.1.102
1A0h	CFG_DATA_LL30_THRESHOLD	CFG_DATA_LL30_THRESHOLD	Section 14.3.3.1.103
1A4h	CFG_DATA_LL31	CFG_DATA_LL31	Section 14.3.3.1.104
1A8h	CFG_DATA_LL31_LPHDR_VAL	CFG_DATA_LL31_LPHDR_VAL	Section 14.3.3.1.105
1ACh	CFG_DATA_LL31_THRESHOLD	CFG_DATA_LL31_THRESHOLD	Section 14.3.3.1.106
1B0h	CFG_LVDS_MAPPING_LANE0_FMT_0	CFG_LVDS_MAPPING_LANE0_FMT_0	Section 14.3.3.1.107
1B4h	CFG_LVDS_MAPPING_LANE1_FMT_0	CFG_LVDS_MAPPING_LANE1_FMT_0	Section 14.3.3.1.108
1B8h	CFG_LVDS_MAPPING_LANE2_FMT_0	CFG_LVDS_MAPPING_LANE2_FMT_0	Section 14.3.3.1.109
1BCh	CFG_LVDS_MAPPING_LANE3_FMT_0	CFG_LVDS_MAPPING_LANE3_FMT_0	Section 14.3.3.1.110
1C0h	CFG_LVDS_MAPPING_LANE0_FMT_1	CFG_LVDS_MAPPING_LANE0_FMT_1	Section 14.3.3.1.111
1C4h	CFG_LVDS_MAPPING_LANE1_FMT_1	CFG_LVDS_MAPPING_LANE1_FMT_1	Section 14.3.3.1.112
1C8h	CFG_LVDS_MAPPING_LANE2_FMT_1	CFG_LVDS_MAPPING_LANE2_FMT_1	Section 14.3.3.1.113
1CCh	CFG_LVDS_MAPPING_LANE3_FMT_1	CFG_LVDS_MAPPING_LANE3_FMT_1	Section 14.3.3.1.114
1D0h	CFG_LVDS_GEN_0	CFG_LVDS_GEN_0	Section 14.3.3.1.115
1D4h	CFG_LVDS_GEN_1	CFG_LVDS_GEN_1	Section 14.3.3.1.116
1D8h	CFG_LVDS_GEN_2	CFG_LVDS_GEN_2	Section 14.3.3.1.117
1DCh	CFG_MASK_REG0	CFG_MASK_REG0	Section 14.3.3.1.118
1E0h	CFG_MASK_REG1	CFG_MASK_REG1	Section 14.3.3.1.119
1E4h	CFG_MASK_REG2	CFG_MASK_REG2	Section 14.3.3.1.120
1E8h	CFG_MASK_REG3	CFG_MASK_REG3	Section 14.3.3.1.121
1ECh	STAT_CBUFF_REG0	STAT_CBUFF_REG0	Section 14.3.3.1.122
1F0h	STAT_CBUFF_REG1	STAT_CBUFF_REG1	Section 14.3.3.1.123
1F4h	STAT_CBUFF_REG2	STAT_CBUFF_REG2	Section 14.3.3.1.124
1F8h	STAT_CBUFF_REG3	STAT_CBUFF_REG3	Section 14.3.3.1.125
20Ch	CLR_CBUFF_REG0	CLR_CBUFF_REG0	Section 14.3.3.1.126
21Ch	STAT_CBUFF_ECC_REG	STAT_CBUFF_ECC_REG	Section 14.3.3.1.127
220h	MASK_CBUFF_ECC_REG	MASK_CBUFF_ECC_REG	Section 14.3.3.1.128
224h	CLR_CBUFF_ECC_REG	CLR_CBUFF_ECC_REG	Section 14.3.3.1.129
228h	STAT_SAFETY	STAT_SAFETY	Section 14.3.3.1.130
22Ch	MASK_SAFETY	MASK_SAFETY	Section 14.3.3.1.131
230h	CLR_SAFETY	CLR_SAFETY	Section 14.3.3.1.132

Complex bit access types are encoded to fit into small table cells. Table 14-5 shows the codes that are used for access types in this section.

Table 14-5. DSS_CBUFF Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

14.3.3.1.1 CONFIG_REG_0 Register (Offset = 0h) [reset = 0h]

CONFIG_REG_0 is shown in Figure 14-6 and described in Table 14-6.

Return to Summary Table.

Basic Config register

Figure 14-6. CONFIG_REG_0 Register

31	30	29	28	27	26	25	24
RESERVED				cswcrst	RESERVED	CFG_FRAME_START_TRIG	CFG_CHIRP_AVAIL_TRIG
R/W-0h				R/W-0h	R/W-0h	0h	0h
23	22	21	20	19	18	17	16
RESERVED						cvc3en	
R/W-0h						R/W-0h	
15	14	13	12	11	10	9	8
cvc2en		cvc1en		cvc0en		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				CFG_SW_TRIG_EN	cftrigen	CFG_ECC_EN	CFG_1LVDS_0CSI
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-6. CONFIG_REG_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	Reserved
27	cswcrst	R/W	0h	CBUFF controller SW Reset 1 => RESET the CBUFF Controller 0 => RELEASE RESET for CBUFF Controller
26	RESERVED	R/W	0h	Reserved
25	CFG_FRAME_START_TRIG		0h	SW Trigger generation : Write 1 to this bit to generate a Frame Start SW Trigger
24	CFG_CHIRP_AVAIL_TRIG		0h	SW Trigger generation : Write 1 to this bit to generate a Chirp Available SW Trigger
23-18	RESERVED	R/W	0h	Reserved
17-16	cvc3en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 3 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 3 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 3 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 3 is generated at end of Frame
15-14	cvc2en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 2 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 2 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 2 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 2 is generated at end of Frame
13-12	cvc1en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 1 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 1 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 1 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 1 is generated at end of Frame

Table 14-6. CONFIG_REG_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	cvc0en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 0 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 0 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 0 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 0 is generated at end of Frame
9-4	RESERVED	R	0h	Reserved
3	CFG_SW_TRIG_EN	R/W	0h	Select Chirp Available Trigger Source 0 : Chirp Available trigger will be generated by HW 1 : Chirp Available trigger will be generated by SW
2	cftrigen	R/W	0h	Select Frame Start Trigger Source 0 : Frame trigger will be generated by HW 1 : Frame trigger will be generated by SW
1	CFG_ECC_EN	R/W	0h	0 : Disable ECC on the CBUF FIFO 1 : Enable ECC on the CBUF FIFO
0	CFG_1LVDS_0CSI	R/W	0h	0 : Send data over CSI-2 1 : Send data over LVDS

14.3.3.1.2 CFG_SPHDR_ADDRESS Register (Offset = 4h) [reset = 0h]

CFG_SPHDR_ADDRESS is shown in Figure 14-7 and described in Table 14-7.

Return to Summary Table.

Short Packet Header Address

Figure 14-7. CFG_SPHDR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_SPHDR_ADDRESS																															
R/W-0h																															

Table 14-7. CFG_SPHDR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_SPHDR_ADDRESS	R/W	0h	CSI2 Programming : Configure the CSI_PROTOCOL_ENGINE__CSI_VC_SHORT_PACKET_HEADER Address in the CSI Protocol Engine LVDS Programming : Configure with the static value : 0x55555555

14.3.3.1.3 CFG_CMD_HSVAL Register (Offset = 8h) [reset = 0h]

CFG_CMD_HSVAL is shown in Figure 14-8 and described in Table 14-8.

Return to Summary Table.

HSYNC Value

Figure 14-8. CFG_CMD_HSVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_CMD_HSVAL																															
R/W-0h																															

Table 14-8. CFG_CMD_HSVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_HSVAL	R/W	0h	CSI2 Programming : Configure the HSync Start Short Packet Value LVDS Programming : If LVDS CRC is enabled : Configure with the static value : 0x55555555 If LVDS CRC is disbaled : Configure with the static value : 0xAAAAAAAA

14.3.3.1.4 CFG_CMD_HEVAL Register (Offset = Ch) [reset = 0h]

CFG_CMD_HEVAL is shown in Figure 14-9 and described in Table 14-9.

Return to Summary Table.

HEND Value

Figure 14-9. CFG_CMD_HEVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_CMD_HEVAL																															
R/W-0h																															

Table 14-9. CFG_CMD_HEVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_HEVAL	R/W	0h	CSI2 Programming : Configure the HSync End Short Packet Value LVDS Programming : If LVDS CRC is enabled : Configure with the static value : 0x33333333 If LVDS CRC is disbaled : Configure with the static value : 0xAAAAAAAA

14.3.3.1.5 CFG_CMD_VSVAL Register (Offset = 10h) [reset = 0h]

CFG_CMD_VSVAL is shown in Figure 14-10 and described in Table 14-10.

Return to Summary Table.

VSYNC Value

Figure 14-10. CFG_CMD_VSVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_CMD_VSVAL																															
R/W-0h																															

Table 14-10. CFG_CMD_VSVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_VSVAL	R/W	0h	CSI2 Programming : Configure the VSync Start Short Packet Value LVDS Programming : Configure with the static value : 0xAAAAAAAA

14.3.3.1.6 CFG_CMD_VEVAL Register (Offset = 14h) [reset = 0h]

CFG_CMD_VEVAL is shown in Figure 14-11 and described in Table 14-11.

Return to Summary Table.

VEND Value

Figure 14-11. CFG_CMD_VEVAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_CMD_VEVAL																															
R/W-0h																															

Table 14-11. CFG_CMD_VEVAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_VEVAL	R/W	0h	CSI2 Programming : Configure the VSync End Short Packet Value LVDS Programming : Configure with the static value : 0xAAAAAAAA

14.3.3.1.7 CFG_LPHDR_ADDRESS Register (Offset = 18h) [reset = 0h]

CFG_LPHDR_ADDRESS is shown in Figure 14-12 and described in Table 14-12.

Return to Summary Table.

Long Packet Address

Figure 14-12. CFG_LPHDR_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_LPHDR_ADDRESS																															
R/W-0h																															

Table 14-12. CFG_LPHDR_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_LPHDR_ADDRESS	R/W	0h	CSI2 Programming : Configure the CSI_PROTOCOL_ENGINE__CSI_VC_LONG_PACKET_HEADER Address in the CSI Protocol Engine LVDS Programming : Configure with the static value : 0x55555555

14.3.3.1.8 CFG_CHIRPS_PER_FRAME Register (Offset = 20h) [reset = 0h]

CFG_CHIRPS_PER_FRAME is shown in Figure 14-13 and described in Table 14-13.

Return to Summary Table.

Number of Chirps per Frame

Figure 14-13. CFG_CHIRPS_PER_FRAME Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_CHIRPS_PER_FRAME																															
R/W-0h																															

Table 14-13. CFG_CHIRPS_PER_FRAME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_CHIRPS_PER_FRAME	R/W	0h	Configure the number of Chirps in a Frame

14.3.3.1.9 CFG_FIFO_FREE_THRESHOLD Register (Offset = 24h) [reset = 01010101h]

CFG_FIFO_FREE_THRESHOLD is shown in Figure 14-14 and described in Table 14-14.

Return to Summary Table.

CSI2 FIFO threshold for transferring data from CBUFF to CSI2

Figure 14-14. CFG_FIFO_FREE_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-1h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_FIFO_FREE_THRESHOLD0							
R/W-1h								R/W-1h							

Table 14-14. CFG_FIFO_FREE_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	1h	Reserved
7-0	CFG_FIFO_FREE_THRESHOLD0	R/W	1h	CSI2 only Programming : Configure the threshold used to fill the FIFO0 in the CSI Protocol engine. CBUFF will send data to the Protocol Engine only if there is a larger number of Free slots that that configured in this register

14.3.3.1.10 CFG_LPPYLD_ADDRESS Register (Offset = 28h) [reset = 0h]

CFG_LPPYLD_ADDRESS is shown in Figure 14-15 and described in Table 14-15.

Return to Summary Table.

Long payload Address

Figure 14-15. CFG_LPPYLD_ADDRESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_LPPYLD_ADDRESS																															
R/W-0h																															

Table 14-15. CFG_LPPYLD_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_LPPYLD_ADDRESS	R/W	0h	CSI2 only Programming : Configure the CSI_PROTOCOL_ENGINE__CSI_VC_LONG_PACKET_PAYLOAD Address in the CSI Protocol Engine

14.3.3.1.11 CFG_DATA_LL0 Register (Offset = 30h) [reset = 0h]

CFG_DATA_LL0 is shown in Figure 14-16 and described in Table 14-16.

Return to Summary Table.

Payload Description : Linked list entry 0

Figure 14-16. CFG_DATA_LL0 Register

31	30	29	28	27	26	25	24
RESERVED			LL0_CRC_EN	LL0_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL0_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL0_SIZE							LL0_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL0_FMT_MAP	LL0_FMT		LL0_VCNUM		LL0_HS	LL0_HE	LL0_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-16. CFG_DATA_LL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL0_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL0_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL0_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL0_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL0_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL0_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL0_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL0_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL0_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL0_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.12 CFG_DATA_LL0_LPHDR_VAL Register (Offset = 34h) [reset = 0h]

CFG_DATA_LL0_LPHDR_VAL is shown in Figure 14-17 and described in Table 14-17.

Return to Summary Table.

Payload Description : Linked list entry 0

Figure 14-17. CFG_DATA_LL0_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL0_LPHDR_VAL																															
R/W-0h																															

Table 14-17. CFG_DATA_LL0_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL0_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.13 CFG_DATA_LL0_THRESHOLD Register (Offset = 38h) [reset = 3F00h]

CFG_DATA_LL0_THRESHOLD is shown in Figure 14-18 and described in Table 14-18.

Return to Summary Table.

Figure 14-18. CFG_DATA_LL0_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll0dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL0_WR_THRESHOLD							NU1	LL0_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-18. CFG_DATA_LL0_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll0dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL0_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL0_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.14 CFG_DATA_LL1 Register (Offset = 3Ch) [reset = 0h]

CFG_DATA_LL1 is shown in Figure 14-19 and described in Table 14-19.

Return to Summary Table.

Figure 14-19. CFG_DATA_LL1 Register

31	30	29	28	27	26	25	24
RESERVED			LL1_CRC_EN	LL1_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL1_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL1_SIZE							LL1_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL1_FMT_MAP	LL1_FMT		LL1_VCNUM		LL1_HS	LL1_HE	LL1_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-19. CFG_DATA_LL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL1_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL1_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL1_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL1_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL1_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL1_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL1_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL1_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL1_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL1_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.15 CFG_DATA_LL1_LPHDR_VAL Register (Offset = 40h) [reset = 0h]

CFG_DATA_LL1_LPHDR_VAL is shown in Figure 14-20 and described in Table 14-20.

Return to Summary Table.

Figure 14-20. CFG_DATA_LL1_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL1_LPHDR_VAL																															
R/W-0h																															

Table 14-20. CFG_DATA_LL1_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL1_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.16 CFG_DATA_LL1_THRESHOLD Register (Offset = 44h) [reset = 3F00h]

CFG_DATA_LL1_THRESHOLD is shown in Figure 14-21 and described in Table 14-21.

Return to Summary Table.

Figure 14-21. CFG_DATA_LL1_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll1dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL1_WR_THRESHOLD							NU1	LL1_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-21. CFG_DATA_LL1_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll1dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL1_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL1_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.17 CFG_DATA_LL2 Register (Offset = 48h) [reset = 0h]

CFG_DATA_LL2 is shown in Figure 14-22 and described in Table 14-22.

Return to Summary Table.

Figure 14-22. CFG_DATA_LL2 Register

31	30	29	28	27	26	25	24
RESERVED			LL2_CRC_EN	LL2_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL2_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL2_SIZE							LL2_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL2_FMT_MAP	LL2_FMT		LL2_VCNUM		LL2_HS	LL2_HE	LL2_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-22. CFG_DATA_LL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL2_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL2_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL2_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL2_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL2_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL2_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL2_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL2_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL2_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL2_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.18 CFG_DATA_LL2_LPHDR_VAL Register (Offset = 4Ch) [reset = 0h]

CFG_DATA_LL2_LPHDR_VAL is shown in Figure 14-23 and described in Table 14-23.

Return to Summary Table.

Figure 14-23. CFG_DATA_LL2_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL2_LPHDR_VAL																															
R/W-0h																															

Table 14-23. CFG_DATA_LL2_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL2_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.19 CFG_DATA_LL2_THRESHOLD Register (Offset = 50h) [reset = 3F00h]

CFG_DATA_LL2_THRESHOLD is shown in Figure 14-24 and described in Table 14-24.

Return to Summary Table.

Figure 14-24. CFG_DATA_LL2_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll2dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL2_WR_THRESHOLD							NU1	LL2_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-24. CFG_DATA_LL2_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll2dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL2_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL2_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.20 CFG_DATA_LL3 Register (Offset = 54h) [reset = 0h]

CFG_DATA_LL3 is shown in Figure 14-25 and described in Table 14-25.

Return to Summary Table.

Figure 14-25. CFG_DATA_LL3 Register

31	30	29	28	27	26	25	24
RESERVED			LL3_CRC_EN	LL3_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL3_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL3_SIZE							LL3_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL3_FMT_MAP	LL3_FMT		LL3_VCNUM		LL3_HS	LL3_HE	LL3_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-25. CFG_DATA_LL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL3_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL3_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL3_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL3_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL3_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL3_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL3_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL3_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL3_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL3_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.21 CFG_DATA_LL3_LPHDR_VAL Register (Offset = 58h) [reset = 0h]

CFG_DATA_LL3_LPHDR_VAL is shown in Figure 14-26 and described in Table 14-26.

Return to Summary Table.

Figure 14-26. CFG_DATA_LL3_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL3_LPHDR_VAL																															
R/W-0h																															

Table 14-26. CFG_DATA_LL3_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL3_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.22 CFG_DATA_LL3_THRESHOLD Register (Offset = 5Ch) [reset = 3F00h]

CFG_DATA_LL3_THRESHOLD is shown in Figure 14-27 and described in Table 14-27.

Return to Summary Table.

Figure 14-27. CFG_DATA_LL3_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll3dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL3_WR_THRESHOLD							NU1	LL3_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-27. CFG_DATA_LL3_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll3dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL3_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL3_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.23 CFG_DATA_LL4 Register (Offset = 60h) [reset = 0h]

CFG_DATA_LL4 is shown in Figure 14-28 and described in Table 14-28.

Return to Summary Table.

Figure 14-28. CFG_DATA_LL4 Register

31	30	29	28	27	26	25	24
RESERVED			LL4_CRC_EN	LL4_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL4_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL4_SIZE							LL4_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL4_FMT_MAP	LL4_FMT		LL4_VCNUM		LL4_HS	LL4_HE	LL4_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-28. CFG_DATA_LL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL4_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL4_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL4_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL4_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL4_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL4_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL4_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL4_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL4_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL4_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.24 CFG_DATA_LL4_LPHDR_VAL Register (Offset = 64h) [reset = 0h]

CFG_DATA_LL4_LPHDR_VAL is shown in Figure 14-29 and described in Table 14-29.

Return to Summary Table.

Figure 14-29. CFG_DATA_LL4_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL4_LPHDR_VAL																															
R/W-0h																															

Table 14-29. CFG_DATA_LL4_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL4_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.25 CFG_DATA_LL4_THRESHOLD Register (Offset = 68h) [reset = 3F00h]

CFG_DATA_LL4_THRESHOLD is shown in Figure 14-30 and described in Table 14-30.

Return to Summary Table.

Figure 14-30. CFG_DATA_LL4_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll4dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL4_WR_THRESHOLD							NU1	LL4_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-30. CFG_DATA_LL4_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll4dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL4_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL4_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.26 CFG_DATA_LL5 Register (Offset = 6Ch) [reset = 0h]

CFG_DATA_LL5 is shown in Figure 14-31 and described in Table 14-31.

Return to Summary Table.

Figure 14-31. CFG_DATA_LL5 Register

31	30	29	28	27	26	25	24
RESERVED			LL5_CRC_EN	LL5_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL5_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL5_SIZE							LL5_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL5_FMT_MAP	LL5_FMT		LL5_VCNUM		LL5_HS	LL5_HE	LL5_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-31. CFG_DATA_LL5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL5_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL5_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL5_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL5_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL5_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL5_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL5_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL5_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL5_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL5_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.27 CFG_DATA_LL5_LPHDR_VAL Register (Offset = 70h) [reset = 0h]

CFG_DATA_LL5_LPHDR_VAL is shown in Figure 14-32 and described in Table 14-32.

Return to Summary Table.

Figure 14-32. CFG_DATA_LL5_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL5_LPHDR_VAL																															
R/W-0h																															

Table 14-32. CFG_DATA_LL5_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL5_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.28 CFG_DATA_LL5_THRESHOLD Register (Offset = 74h) [reset = 3F00h]

CFG_DATA_LL5_THRESHOLD is shown in Figure 14-33 and described in Table 14-33.

Return to Summary Table.

Figure 14-33. CFG_DATA_LL5_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll5dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL5_WR_THRESHOLD							NU1	LL5_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-33. CFG_DATA_LL5_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll5dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL5_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL5_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.29 CFG_DATA_LL6 Register (Offset = 78h) [reset = 0h]

CFG_DATA_LL6 is shown in Figure 14-34 and described in Table 14-34.

Return to Summary Table.

Figure 14-34. CFG_DATA_LL6 Register

31	30	29	28	27	26	25	24
RESERVED			LL6_CRC_EN	LL6_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL6_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL6_SIZE							LL6_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL6_FMT_MAP	LL6_FMT		LL6_VCNUM		LL6_HS	LL6_HE	LL6_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-34. CFG_DATA_LL6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL6_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL6_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL6_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL6_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL6_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL6_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL6_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL6_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL6_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL6_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.30 CFG_DATA_LL6_LPHDR_VAL Register (Offset = 7Ch) [reset = 0h]

CFG_DATA_LL6_LPHDR_VAL is shown in Figure 14-35 and described in Table 14-35.

Return to Summary Table.

Figure 14-35. CFG_DATA_LL6_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL6_LPHDR_VAL																															
R/W-0h																															

Table 14-35. CFG_DATA_LL6_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL6_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.31 CFG_DATA_LL6_THRESHOLD Register (Offset = 80h) [reset = 3F00h]

CFG_DATA_LL6_THRESHOLD is shown in Figure 14-36 and described in Table 14-36.

Return to Summary Table.

Figure 14-36. CFG_DATA_LL6_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll6dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL6_WR_THRESHOLD							NU1	LL6_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-36. CFG_DATA_LL6_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll6dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL6_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL6_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.32 CFG_DATA_LL7 Register (Offset = 84h) [reset = 0h]

CFG_DATA_LL7 is shown in Figure 14-37 and described in Table 14-37.

Return to Summary Table.

Figure 14-37. CFG_DATA_LL7 Register

31	30	29	28	27	26	25	24
RESERVED			LL7_CRC_EN	LL7_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL7_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL7_SIZE							LL7_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL7_FMT_MAP	LL7_FMT		LL7_VCNUM		LL7_HS	LL7_HE	LL7_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-37. CFG_DATA_LL7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL7_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL7_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL7_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL7_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL7_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL7_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL7_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL7_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL7_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL7_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.33 CFG_DATA_LL7_LPHDR_VAL Register (Offset = 88h) [reset = 0h]

CFG_DATA_LL7_LPHDR_VAL is shown in Figure 14-38 and described in Table 14-38.

Return to Summary Table.

Figure 14-38. CFG_DATA_LL7_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL7_LPHDR_VAL																															
R/W-0h																															

Table 14-38. CFG_DATA_LL7_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL7_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.34 CFG_DATA_LL7_THRESHOLD Register (Offset = 8Ch) [reset = 3F00h]

CFG_DATA_LL7_THRESHOLD is shown in Figure 14-39 and described in Table 14-39.

Return to Summary Table.

Figure 14-39. CFG_DATA_LL7_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll7dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL7_WR_THRESHOLD							NU1	LL7_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-39. CFG_DATA_LL7_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll7dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL7_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL7_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.35 CFG_DATA_LL8 Register (Offset = 90h) [reset = 0h]

CFG_DATA_LL8 is shown in Figure 14-40 and described in Table 14-40.

Return to Summary Table.

Figure 14-40. CFG_DATA_LL8 Register

31	30	29	28	27	26	25	24
RESERVED			LL8_CRC_EN	LL8_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL8_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL8_SIZE							LL8_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL8_FMT_MAP	LL8_FMT		LL8_VCNUM		LL8_HS	LL8_HE	LL8_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-40. CFG_DATA_LL8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL8_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL8_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL8_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL8_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL8_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL8_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL8_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL8_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL8_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL8_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.36 CFG_DATA_LL8_LPHDR_VAL Register (Offset = 94h) [reset = 0h]

CFG_DATA_LL8_LPHDR_VAL is shown in Figure 14-41 and described in Table 14-41.

Return to Summary Table.

Figure 14-41. CFG_DATA_LL8_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL8_LPHDR_VAL																															
R/W-0h																															

Table 14-41. CFG_DATA_LL8_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL8_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.37 CFG_DATA_LL8_THRESHOLD Register (Offset = 98h) [reset = 3F00h]

CFG_DATA_LL8_THRESHOLD is shown in Figure 14-42 and described in Table 14-42.

Return to Summary Table.

Figure 14-42. CFG_DATA_LL8_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll8dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL8_WR_THRESHOLD							NU1	LL8_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-42. CFG_DATA_LL8_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll8dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL8_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL8_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.38 CFG_DATA_LL9 Register (Offset = 9Ch) [reset = 0h]

CFG_DATA_LL9 is shown in Figure 14-43 and described in Table 14-43.

Return to Summary Table.

Figure 14-43. CFG_DATA_LL9 Register

31	30	29	28	27	26	25	24
RESERVED			LL9_CRC_EN	LL9_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL9_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL9_SIZE							LL9_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL9_FMT_MAP	LL9_FMT		LL9_VCNUM		LL9_HS	LL9_HE	LL9_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-43. CFG_DATA_LL9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL9_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL9_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL9_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL9_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL9_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL9_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL9_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL9_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL9_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL9_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.39 CFG_DATA_LL9_LPHDR_VAL Register (Offset = A0h) [reset = 0h]

CFG_DATA_LL9_LPHDR_VAL is shown in Figure 14-44 and described in Table 14-44.

Return to Summary Table.

Figure 14-44. CFG_DATA_LL9_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL9_LPHDR_VAL																															
R/W-0h																															

Table 14-44. CFG_DATA_LL9_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL9_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.40 CFG_DATA_LL9_THRESHOLD Register (Offset = A4h) [reset = 3F00h]

CFG_DATA_LL9_THRESHOLD is shown in Figure 14-45 and described in Table 14-45.

Return to Summary Table.

Figure 14-45. CFG_DATA_LL9_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll9dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL9_WR_THRESHOLD							NU1	LL9_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-45. CFG_DATA_LL9_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll9dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL9_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL9_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.41 CFG_DATA_LL10 Register (Offset = A8h) [reset = 0h]

CFG_DATA_LL10 is shown in Figure 14-46 and described in Table 14-46.

Return to Summary Table.

Figure 14-46. CFG_DATA_LL10 Register

31	30	29	28	27	26	25	24
RESERVED			LL10_CRC_EN	LL10_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL10_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL10_SIZE							LL10_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL10_FMT_MAP	LL10_FMT		LL10_VCNUM		LL10_HS	LL10_HE	LL10_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-46. CFG_DATA_LL10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL10_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL10_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL10_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL10_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL10_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL10_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL10_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL10_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL10_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL10_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.42 CFG_DATA_LL10_LPHDR_VAL Register (Offset = ACh) [reset = 0h]

CFG_DATA_LL10_LPHDR_VAL is shown in Figure 14-47 and described in Table 14-47.

Return to Summary Table.

Figure 14-47. CFG_DATA_LL10_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL10_LPHDR_VAL																															
R/W-0h																															

Table 14-47. CFG_DATA_LL10_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL10_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.43 CFG_DATA_LL10_THRESHOLD Register (Offset = B0h) [reset = 3F00h]

CFG_DATA_LL10_THRESHOLD is shown in Figure 14-48 and described in Table 14-48.

Return to Summary Table.

Figure 14-48. CFG_DATA_LL10_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll10dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL10_WR_THRESHOLD							NU1	LL10_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-48. CFG_DATA_LL10_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll10dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL10_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL10_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.44 CFG_DATA_LL11 Register (Offset = B4h) [reset = 0h]

CFG_DATA_LL11 is shown in Figure 14-49 and described in Table 14-49.

Return to Summary Table.

Figure 14-49. CFG_DATA_LL11 Register

31	30	29	28	27	26	25	24
RESERVED			LL11_CRC_EN	LL11_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL11_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL11_SIZE							LL11_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL11_FMT_MAP	LL11_FMT		LL11_VCNUM		LL11_HS	LL11_HE	LL11_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-49. CFG_DATA_LL11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL11_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL11_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL11_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL11_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL11_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL11_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL11_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL11_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL11_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL11_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.45 CFG_DATA_LL11_LPHDR_VAL Register (Offset = B8h) [reset = 0h]

CFG_DATA_LL11_LPHDR_VAL is shown in Figure 14-50 and described in Table 14-50.

Return to Summary Table.

Figure 14-50. CFG_DATA_LL11_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL11_LPHDR_VAL																															
R/W-0h																															

Table 14-50. CFG_DATA_LL11_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL11_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.46 CFG_DATA_LL11_THRESHOLD Register (Offset = BCh) [reset = 3F00h]

CFG_DATA_LL11_THRESHOLD is shown in Figure 14-51 and described in Table 14-51.

Return to Summary Table.

Figure 14-51. CFG_DATA_LL11_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll11dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL11_WR_THRESHOLD							NU1	LL11_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-51. CFG_DATA_LL11_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll11dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL11_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL11_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.47 CFG_DATA_LL12 Register (Offset = C0h) [reset = 0h]

CFG_DATA_LL12 is shown in Figure 14-52 and described in Table 14-52.

Return to Summary Table.

Figure 14-52. CFG_DATA_LL12 Register

31	30	29	28	27	26	25	24
RESERVED			LL12_CRC_EN	LL12_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL12_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL12_SIZE							LL12_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL12_FMT_MAP	LL12_FMT		LL12_VCNUM		LL12_HS	LL12_HE	LL12_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-52. CFG_DATA_LL12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL12_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL12_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL12_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL12_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL12_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL12_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL12_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL12_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL12_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL12_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.48 CFG_DATA_LL12_LPHDR_VAL Register (Offset = C4h) [reset = 0h]

CFG_DATA_LL12_LPHDR_VAL is shown in Figure 14-53 and described in Table 14-53.

Return to Summary Table.

Figure 14-53. CFG_DATA_LL12_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL12_LPHDR_VAL																															
R/W-0h																															

Table 14-53. CFG_DATA_LL12_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL12_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.49 CFG_DATA_LL12_THRESHOLD Register (Offset = C8h) [reset = 3F00h]

CFG_DATA_LL12_THRESHOLD is shown in Figure 14-54 and described in Table 14-54.

Return to Summary Table.

Figure 14-54. CFG_DATA_LL12_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll12dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL12_WR_THRESHOLD							NU1	LL12_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-54. CFG_DATA_LL12_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll12dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL12_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL12_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.50 CFG_DATA_LL13 Register (Offset = CCh) [reset = 0h]

CFG_DATA_LL13 is shown in Figure 14-55 and described in Table 14-55.

Return to Summary Table.

Figure 14-55. CFG_DATA_LL13 Register

31	30	29	28	27	26	25	24
RESERVED			LL13_CRC_EN	LL13_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL13_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL13_SIZE							LL13_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL13_FMT_MAP	LL13_FMT		LL13_VCNUM		LL13_HS	LL13_HE	LL13_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-55. CFG_DATA_LL13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL13_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL13_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL13_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL13_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL13_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL13_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL13_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL13_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL13_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL13_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.51 CFG_DATA_LL13_LPHDR_VAL Register (Offset = D0h) [reset = 0h]

CFG_DATA_LL13_LPHDR_VAL is shown in Figure 14-56 and described in Table 14-56.

Return to Summary Table.

Figure 14-56. CFG_DATA_LL13_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL13_LPHDR_VAL																															
R/W-0h																															

Table 14-56. CFG_DATA_LL13_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL13_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.52 CFG_DATA_LL13_THRESHOLD Register (Offset = D4h) [reset = 3F00h]

CFG_DATA_LL13_THRESHOLD is shown in Figure 14-57 and described in Table 14-57.

Return to Summary Table.

Figure 14-57. CFG_DATA_LL13_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll13dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL13_WR_THRESHOLD							NU1	LL13_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-57. CFG_DATA_LL13_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll13dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL13_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL13_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.53 CFG_DATA_LL14 Register (Offset = D8h) [reset = 0h]

CFG_DATA_LL14 is shown in Figure 14-58 and described in Table 14-58.

Return to Summary Table.

Figure 14-58. CFG_DATA_LL14 Register

31	30	29	28	27	26	25	24
RESERVED			LL14_CRC_EN	LL14_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL14_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL14_SIZE							LL14_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL14_FMT_MAP	LL14_FMT		LL14_VCNUM		LL14_HS	LL14_HE	LL14_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-58. CFG_DATA_LL14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL14_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL14_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL14_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL14_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL14_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL14_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL14_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL14_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL14_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL14_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.54 CFG_DATA_LL14_LPHDR_VAL Register (Offset = DCh) [reset = 0h]

CFG_DATA_LL14_LPHDR_VAL is shown in Figure 14-59 and described in Table 14-59.

Return to Summary Table.

Figure 14-59. CFG_DATA_LL14_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL14_LPHDR_VAL																															
R/W-0h																															

Table 14-59. CFG_DATA_LL14_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL14_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.55 CFG_DATA_LL14_THRESHOLD Register (Offset = E0h) [reset = 3F00h]

CFG_DATA_LL14_THRESHOLD is shown in Figure 14-60 and described in Table 14-60.

Return to Summary Table.

Figure 14-60. CFG_DATA_LL14_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll14dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL14_WR_THRESHOLD							NU1	LL14_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-60. CFG_DATA_LL14_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll14dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL14_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL14_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.56 CFG_DATA_LL15 Register (Offset = E4h) [reset = 0h]

CFG_DATA_LL15 is shown in Figure 14-61 and described in Table 14-61.

Return to Summary Table.

Figure 14-61. CFG_DATA_LL15 Register

31	30	29	28	27	26	25	24
RESERVED			LL15_CRC_EN	LL15_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL15_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL15_SIZE							LL15_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL15_FMT_MAP	LL15_FMT		LL15_VCNUM		LL15_HS	LL15_HE	LL15_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-61. CFG_DATA_LL15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL15_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL15_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL15_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL15_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL15_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL15_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL15_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL15_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL15_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL15_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.57 CFG_DATA_LL15_LPHDR_VAL Register (Offset = E8h) [reset = 0h]

CFG_DATA_LL15_LPHDR_VAL is shown in Figure 14-62 and described in Table 14-62.

Return to Summary Table.

Figure 14-62. CFG_DATA_LL15_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL15_LPHDR_VAL																															
R/W-0h																															

Table 14-62. CFG_DATA_LL15_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL15_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.58 CFG_DATA_LL15_THRESHOLD Register (Offset = ECh) [reset = 3F00h]

CFG_DATA_LL15_THRESHOLD is shown in Figure 14-63 and described in Table 14-63.

Return to Summary Table.

Figure 14-63. CFG_DATA_LL15_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll15dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL15_WR_THRESHOLD							NU1	LL15_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-63. CFG_DATA_LL15_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll15dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL15_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL15_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.59 CFG_DATA_LL16 Register (Offset = F0h) [reset = 0h]

CFG_DATA_LL16 is shown in Figure 14-64 and described in Table 14-64.

Return to Summary Table.

Figure 14-64. CFG_DATA_LL16 Register

31	30	29	28	27	26	25	24
RESERVED			LL16_CRC_EN	LL16_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL16_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL16_SIZE							LL16_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL16_FMT_MAP	LL16_FMT		LL16_VCNUM		LL16_HS	LL16_HE	LL16_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-64. CFG_DATA_LL16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL16_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL16_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL16_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL16_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL16_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL16_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL16_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL16_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL16_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL16_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.60 CFG_DATA_LL16_LPHDR_VAL Register (Offset = F4h) [reset = 0h]

CFG_DATA_LL16_LPHDR_VAL is shown in Figure 14-65 and described in Table 14-65.

Return to Summary Table.

Figure 14-65. CFG_DATA_LL16_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL16_LPHDR_VAL																															
R/W-0h																															

Table 14-65. CFG_DATA_LL16_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL16_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.61 CFG_DATA_LL16_THRESHOLD Register (Offset = F8h) [reset = 3F00h]

CFG_DATA_LL16_THRESHOLD is shown in Figure 14-66 and described in Table 14-66.

Return to Summary Table.

Figure 14-66. CFG_DATA_LL16_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll16dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL16_WR_THRESHOLD							NU1	LL16_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-66. CFG_DATA_LL16_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll16dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL16_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL16_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.62 CFG_DATA_LL17 Register (Offset = FCh) [reset = 0h]

CFG_DATA_LL17 is shown in Figure 14-67 and described in Table 14-67.

Return to Summary Table.

Figure 14-67. CFG_DATA_LL17 Register

31	30	29	28	27	26	25	24
RESERVED			LL17_CRC_EN	LL17_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL17_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL17_SIZE							LL17_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL17_FMT_MAP	LL17_FMT		LL17_VCNUM		LL17_HS	LL17_HE	LL17_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-67. CFG_DATA_LL17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL17_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL17_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL17_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL17_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL17_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL17_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL17_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL17_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL17_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL17_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.63 CFG_DATA_LL17_LPHDR_VAL Register (Offset = 100h) [reset = 0h]

CFG_DATA_LL17_LPHDR_VAL is shown in Figure 14-68 and described in Table 14-68.

Return to Summary Table.

Figure 14-68. CFG_DATA_LL17_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL17_LPHDR_VAL																															
R/W-0h																															

Table 14-68. CFG_DATA_LL17_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL17_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.64 CFG_DATA_LL17_THRESHOLD Register (Offset = 104h) [reset = 3F00h]

CFG_DATA_LL17_THRESHOLD is shown in Figure 14-69 and described in Table 14-69.

Return to Summary Table.

Figure 14-69. CFG_DATA_LL17_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll17dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL17_WR_THRESHOLD							NU1	LL17_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-69. CFG_DATA_LL17_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll17dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL17_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL17_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.65 CFG_DATA_LL18 Register (Offset = 108h) [reset = 0h]

CFG_DATA_LL18 is shown in Figure 14-70 and described in Table 14-70.

Return to Summary Table.

Figure 14-70. CFG_DATA_LL18 Register

31	30	29	28	27	26	25	24
RESERVED			LL18_CRC_EN	LL18_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL18_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL18_SIZE							LL18_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL18_FMT_MAP	LL18_FMT		LL18_VCNUM		LL18_HS	LL18_HE	LL18_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-70. CFG_DATA_LL18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL18_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL18_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL18_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL18_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL18_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL18_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL18_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL18_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL18_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL18_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.66 CFG_DATA_LL18_LPHDR_VAL Register (Offset = 10Ch) [reset = 0h]

CFG_DATA_LL18_LPHDR_VAL is shown in Figure 14-71 and described in Table 14-71.

Return to Summary Table.

Figure 14-71. CFG_DATA_LL18_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL18_LPHDR_VAL																															
R/W-0h																															

Table 14-71. CFG_DATA_LL18_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL18_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.67 CFG_DATA_LL18_THRESHOLD Register (Offset = 110h) [reset = 3F00h]

CFG_DATA_LL18_THRESHOLD is shown in Figure 14-72 and described in Table 14-72.

Return to Summary Table.

Figure 14-72. CFG_DATA_LL18_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll18dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL18_WR_THRESHOLD							NU1	LL18_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-72. CFG_DATA_LL18_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll18dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL18_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL18_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.68 CFG_DATA_LL19 Register (Offset = 114h) [reset = 0h]

CFG_DATA_LL19 is shown in Figure 14-73 and described in Table 14-73.

Return to Summary Table.

Figure 14-73. CFG_DATA_LL19 Register

31	30	29	28	27	26	25	24
RESERVED			LL19_CRC_EN	LL19_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL19_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL19_SIZE							LL19_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL19_FMT_MAP	LL19_FMT		LL19_VCNUM		LL19_HS	LL19_HE	LL19_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-73. CFG_DATA_LL19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL19_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL19_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL19_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL19_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL19_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL19_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL19_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL19_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL19_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL19_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.69 CFG_DATA_LL19_LPHDR_VAL Register (Offset = 118h) [reset = 0h]

CFG_DATA_LL19_LPHDR_VAL is shown in Figure 14-74 and described in Table 14-74.

Return to Summary Table.

Figure 14-74. CFG_DATA_LL19_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL19_LPHDR_VAL																															
R/W-0h																															

Table 14-74. CFG_DATA_LL19_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL19_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.70 CFG_DATA_LL19_THRESHOLD Register (Offset = 11Ch) [reset = 3F00h]

CFG_DATA_LL19_THRESHOLD is shown in Figure 14-75 and described in Table 14-75.

Return to Summary Table.

Figure 14-75. CFG_DATA_LL19_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll19dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL19_WR_THRESHOLD							NU1	LL19_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-75. CFG_DATA_LL19_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll19dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL19_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL19_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.71 CFG_DATA_LL20 Register (Offset = 120h) [reset = 0h]

CFG_DATA_LL20 is shown in Figure 14-76 and described in Table 14-76.

Return to Summary Table.

Figure 14-76. CFG_DATA_LL20 Register

31	30	29	28	27	26	25	24
RESERVED			LL20_CRC_EN	LL20_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL20_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL20_SIZE							LL20_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL20_FMT_MAP	LL20_FMT		LL20_VCNUM		LL20_HS	LL20_HE	LL20_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-76. CFG_DATA_LL20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL20_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL20_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL20_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL20_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL20_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL20_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL20_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL20_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL20_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL20_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.72 CFG_DATA_LL20_LPHDR_VAL Register (Offset = 124h) [reset = 0h]

CFG_DATA_LL20_LPHDR_VAL is shown in Figure 14-77 and described in Table 14-77.

Return to Summary Table.

Figure 14-77. CFG_DATA_LL20_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL20_LPHDR_VAL																															
R/W-0h																															

Table 14-77. CFG_DATA_LL20_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL20_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.73 CFG_DATA_LL20_THRESHOLD Register (Offset = 128h) [reset = 3F00h]

CFG_DATA_LL20_THRESHOLD is shown in Figure 14-78 and described in Table 14-78.

Return to Summary Table.

Figure 14-78. CFG_DATA_LL20_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll20dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL20_WR_THRESHOLD							NU1	LL20_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-78. CFG_DATA_LL20_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll20dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL20_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL20_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.74 CFG_DATA_LL21 Register (Offset = 12Ch) [reset = 0h]

CFG_DATA_LL21 is shown in Figure 14-79 and described in Table 14-79.

Return to Summary Table.

Figure 14-79. CFG_DATA_LL21 Register

31	30	29	28	27	26	25	24
RESERVED			LL21_CRC_EN	LL21_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL21_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL21_SIZE							LL21_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL21_FMT_MAP	LL21_FMT		LL21_VCNUM		LL21_HS	LL21_HE	LL21_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-79. CFG_DATA_LL21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL21_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL21_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL21_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL21_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL21_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL21_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL21_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL21_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL21_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL21_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.75 CFG_DATA_LL21_LPHDR_VAL Register (Offset = 130h) [reset = 0h]

CFG_DATA_LL21_LPHDR_VAL is shown in Figure 14-80 and described in Table 14-80.

Return to Summary Table.

Figure 14-80. CFG_DATA_LL21_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL21_LPHDR_VAL																															
R/W-0h																															

Table 14-80. CFG_DATA_LL21_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL21_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.76 CFG_DATA_LL21_THRESHOLD Register (Offset = 134h) [reset = 3F00h]

CFG_DATA_LL21_THRESHOLD is shown in Figure 14-81 and described in Table 14-81.

Return to Summary Table.

Figure 14-81. CFG_DATA_LL21_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll21dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL21_WR_THRESHOLD							NU1	LL21_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-81. CFG_DATA_LL21_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll21dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL21_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL21_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.77 CFG_DATA_LL22 Register (Offset = 138h) [reset = 0h]

CFG_DATA_LL22 is shown in Figure 14-82 and described in Table 14-82.

Return to Summary Table.

Figure 14-82. CFG_DATA_LL22 Register

31	30	29	28	27	26	25	24
RESERVED			LL22_CRC_EN	LL22_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL22_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL22_SIZE							LL22_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL22_FMT_MAP	LL22_FMT		LL22_VCNUM		LL22_HS	LL22_HE	LL22_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-82. CFG_DATA_LL22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL22_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL22_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL22_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL22_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL22_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL22_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL22_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL22_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL22_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL22_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.78 CFG_DATA_LL22_LPHDR_VAL Register (Offset = 13Ch) [reset = 0h]

CFG_DATA_LL22_LPHDR_VAL is shown in Figure 14-83 and described in Table 14-83.

Return to Summary Table.

Figure 14-83. CFG_DATA_LL22_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL22_LPHDR_VAL																															
R/W-0h																															

Table 14-83. CFG_DATA_LL22_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL22_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.79 CFG_DATA_LL22_THRESHOLD Register (Offset = 140h) [reset = 3F00h]

CFG_DATA_LL22_THRESHOLD is shown in Figure 14-84 and described in Table 14-84.

Return to Summary Table.

Figure 14-84. CFG_DATA_LL22_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll22dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL22_WR_THRESHOLD							NU1	LL22_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-84. CFG_DATA_LL22_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll22dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL22_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL22_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.80 CFG_DATA_LL23 Register (Offset = 144h) [reset = 0h]

CFG_DATA_LL23 is shown in Figure 14-85 and described in Table 14-85.

Return to Summary Table.

Figure 14-85. CFG_DATA_LL23 Register

31	30	29	28	27	26	25	24
RESERVED			LL23_CRC_EN	LL23_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL23_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL23_SIZE							LL23_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL23_FMT_MAP	LL23_FMT		LL23_VCNUM		LL23_HS	LL23_HE	LL23_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-85. CFG_DATA_LL23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL23_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL23_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL23_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL23_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL23_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL23_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL23_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL23_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL23_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL23_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.81 CFG_DATA_LL23_LPHDR_VAL Register (Offset = 148h) [reset = 0h]

CFG_DATA_LL23_LPHDR_VAL is shown in Figure 14-86 and described in Table 14-86.

Return to Summary Table.

Figure 14-86. CFG_DATA_LL23_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL23_LPHDR_VAL																															
R/W-0h																															

Table 14-86. CFG_DATA_LL23_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL23_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.82 CFG_DATA_LL23_THRESHOLD Register (Offset = 14Ch) [reset = 3F00h]

CFG_DATA_LL23_THRESHOLD is shown in Figure 14-87 and described in Table 14-87.

Return to Summary Table.

Figure 14-87. CFG_DATA_LL23_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll23dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL23_WR_THRESHOLD							NU1	LL23_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-87. CFG_DATA_LL23_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll23dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL23_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL23_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.83 CFG_DATA_LL24 Register (Offset = 150h) [reset = 0h]

CFG_DATA_LL24 is shown in Figure 14-88 and described in Table 14-88.

Return to Summary Table.

Figure 14-88. CFG_DATA_LL24 Register

31	30	29	28	27	26	25	24
RESERVED			LL24_CRC_EN	LL24_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL24_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL24_SIZE							LL24_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL24_FMT_MAP	LL24_FMT		LL24_VCNUM		LL24_HS	LL24_HE	LL24_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-88. CFG_DATA_LL24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL24_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL24_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL24_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL24_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL24_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL24_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL24_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL24_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL24_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL24_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.84 CFG_DATA_LL24_LPHDR_VAL Register (Offset = 154h) [reset = 0h]

CFG_DATA_LL24_LPHDR_VAL is shown in Figure 14-89 and described in Table 14-89.

Return to Summary Table.

Figure 14-89. CFG_DATA_LL24_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL24_LPHDR_VAL																															
R/W-0h																															

Table 14-89. CFG_DATA_LL24_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL24_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.85 CFG_DATA_LL24_THRESHOLD Register (Offset = 158h) [reset = 3F00h]

CFG_DATA_LL24_THRESHOLD is shown in Figure 14-90 and described in Table 14-90.

Return to Summary Table.

Figure 14-90. CFG_DATA_LL24_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll24dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL24_WR_THRESHOLD							NU1	LL24_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-90. CFG_DATA_LL24_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll24dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL24_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL24_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.86 CFG_DATA_LL25 Register (Offset = 15Ch) [reset = 0h]

CFG_DATA_LL25 is shown in Figure 14-91 and described in Table 14-91.

Return to Summary Table.

Figure 14-91. CFG_DATA_LL25 Register

31	30	29	28	27	26	25	24
RESERVED			LL25_CRC_EN	LL25_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL25_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL25_SIZE							LL25_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL25_FMT_MAP	LL25_FMT		LL25_VCNUM		LL25_HS	LL25_HE	LL25_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-91. CFG_DATA_LL25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL25_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL25_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL25_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL25_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL25_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL25_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL25_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL25_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL25_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL25_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.87 CFG_DATA_LL25_LPHDR_VAL Register (Offset = 160h) [reset = 0h]

CFG_DATA_LL25_LPHDR_VAL is shown in Figure 14-92 and described in Table 14-92.

Return to Summary Table.

Figure 14-92. CFG_DATA_LL25_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL25_LPHDR_VAL																															
R/W-0h																															

Table 14-92. CFG_DATA_LL25_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL25_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.88 CFG_DATA_LL25_THRESHOLD Register (Offset = 164h) [reset = 3F00h]

CFG_DATA_LL25_THRESHOLD is shown in Figure 14-93 and described in Table 14-93.

Return to Summary Table.

Figure 14-93. CFG_DATA_LL25_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll25dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL25_WR_THRESHOLD							NU1	LL25_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-93. CFG_DATA_LL25_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll25dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL25_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL25_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.89 CFG_DATA_LL26 Register (Offset = 168h) [reset = 0h]

CFG_DATA_LL26 is shown in Figure 14-94 and described in Table 14-94.

Return to Summary Table.

Figure 14-94. CFG_DATA_LL26 Register

31	30	29	28	27	26	25	24
RESERVED			LL26_CRC_EN	LL26_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL26_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL26_SIZE							LL26_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL26_FMT_MAP	LL26_FMT		LL26_VCNUM		LL26_HS	LL26_HE	LL26_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-94. CFG_DATA_LL26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL26_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL26_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL26_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL26_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL26_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL26_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL26_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL26_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL26_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL26_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.90 CFG_DATA_LL26_LPHDR_VAL Register (Offset = 16Ch) [reset = 0h]

CFG_DATA_LL26_LPHDR_VAL is shown in Figure 14-95 and described in Table 14-95.

Return to Summary Table.

Figure 14-95. CFG_DATA_LL26_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL26_LPHDR_VAL																															
R/W-0h																															

Table 14-95. CFG_DATA_LL26_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL26_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.91 CFG_DATA_LL26_THRESHOLD Register (Offset = 170h) [reset = 3F00h]

CFG_DATA_LL26_THRESHOLD is shown in Figure 14-96 and described in Table 14-96.

Return to Summary Table.

Figure 14-96. CFG_DATA_LL26_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll26dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL26_WR_THRESHOLD							NU1	LL26_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-96. CFG_DATA_LL26_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll26dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL26_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL26_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.92 CFG_DATA_LL27 Register (Offset = 174h) [reset = 0h]

CFG_DATA_LL27 is shown in Figure 14-97 and described in Table 14-97.

Return to Summary Table.

Figure 14-97. CFG_DATA_LL27 Register

31	30	29	28	27	26	25	24
RESERVED			LL27_CRC_EN	LL27_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL27_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL27_SIZE							LL27_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL27_FMT_MAP	LL27_FMT		LL27_VCNUM		LL27_HS	LL27_HE	LL27_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-97. CFG_DATA_LL27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL27_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL27_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL27_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL27_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL27_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL27_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL27_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL27_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL27_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL27_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.93 CFG_DATA_LL27_LPHDR_VAL Register (Offset = 178h) [reset = 0h]

CFG_DATA_LL27_LPHDR_VAL is shown in Figure 14-98 and described in Table 14-98.

Return to Summary Table.

Figure 14-98. CFG_DATA_LL27_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL27_LPHDR_VAL																															
R/W-0h																															

Table 14-98. CFG_DATA_LL27_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL27_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.94 CFG_DATA_LL27_THRESHOLD Register (Offset = 17Ch) [reset = 3F00h]

CFG_DATA_LL27_THRESHOLD is shown in Figure 14-99 and described in Table 14-99.

Return to Summary Table.

Figure 14-99. CFG_DATA_LL27_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll27dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL27_WR_THRESHOLD							NU1	LL27_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-99. CFG_DATA_LL27_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll27dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL27_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL27_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.95 CFG_DATA_LL28 Register (Offset = 180h) [reset = 0h]

CFG_DATA_LL28 is shown in Figure 14-100 and described in Table 14-100.

Return to Summary Table.

Figure 14-100. CFG_DATA_LL28 Register

31	30	29	28	27	26	25	24
RESERVED			LL28_CRC_EN	LL28_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL28_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL28_SIZE							LL28_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL28_FMT_MAP	LL28_FMT		LL28_VCNUM		LL28_HS	LL28_HE	LL28_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-100. CFG_DATA_LL28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL28_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL28_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL28_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL28_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL28_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL28_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL28_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL28_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL28_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL28_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.96 CFG_DATA_LL28_LPHDR_VAL Register (Offset = 184h) [reset = 0h]

CFG_DATA_LL28_LPHDR_VAL is shown in Figure 14-101 and described in Table 14-101.

Return to Summary Table.

Figure 14-101. CFG_DATA_LL28_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL28_LPHDR_VAL																															
R/W-0h																															

Table 14-101. CFG_DATA_LL28_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL28_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.97 CFG_DATA_LL28_THRESHOLD Register (Offset = 188h) [reset = 3F00h]

CFG_DATA_LL28_THRESHOLD is shown in Figure 14-102 and described in Table 14-102.

Return to Summary Table.

Figure 14-102. CFG_DATA_LL28_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll28dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL28_WR_THRESHOLD							NU1	LL28_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-102. CFG_DATA_LL28_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll28dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL28_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL28_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.98 CFG_DATA_LL29 Register (Offset = 18Ch) [reset = 0h]

CFG_DATA_LL29 is shown in Figure 14-103 and described in Table 14-103.

Return to Summary Table.

Figure 14-103. CFG_DATA_LL29 Register

31	30	29	28	27	26	25	24
RESERVED			LL29_CRC_EN	LL29_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL29_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL29_SIZE							LL29_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL29_FMT_MAP	LL29_FMT		LL29_VCNUM		LL29_HS	LL29_HE	LL29_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-103. CFG_DATA_LL29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL29_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL29_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL29_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL29_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL29_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL29_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL29_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL29_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL29_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL29_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.99 CFG_DATA_LL29_LPHDR_VAL Register (Offset = 190h) [reset = 0h]

CFG_DATA_LL29_LPHDR_VAL is shown in Figure 14-104 and described in Table 14-104.

Return to Summary Table.

Figure 14-104. CFG_DATA_LL29_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL29_LPHDR_VAL																															
R/W-0h																															

Table 14-104. CFG_DATA_LL29_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL29_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.100 CFG_DATA_LL29_THRESHOLD Register (Offset = 194h) [reset = 3F00h]

CFG_DATA_LL29_THRESHOLD is shown in Figure 14-105 and described in Table 14-105.

Return to Summary Table.

Figure 14-105. CFG_DATA_LL29_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll29dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL29_WR_THRESHOLD							NU1	LL29_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-105. CFG_DATA_LL29_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll29dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL29_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL29_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.101 CFG_DATA_LL30 Register (Offset = 198h) [reset = 0h]

CFG_DATA_LL30 is shown in Figure 14-106 and described in Table 14-106.

Return to Summary Table.

Figure 14-106. CFG_DATA_LL30 Register

31	30	29	28	27	26	25	24
RESERVED			LL30_CRC_EN	LL30_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL30_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL30_SIZE							LL30_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL30_FMT_MAP	LL30_FMT		LL30_VCNUM		LL30_HS	LL30_HE	LL30_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-106. CFG_DATA_LL30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL30_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL30_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL30_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL30_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL30_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL30_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL30_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL30_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL30_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL30_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.102 CFG_DATA_LL30_LPHDR_VAL Register (Offset = 19Ch) [reset = 0h]

CFG_DATA_LL30_LPHDR_VAL is shown in Figure 14-107 and described in Table 14-107.

Return to Summary Table.

Figure 14-107. CFG_DATA_LL30_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL30_LPHDR_VAL																															
R/W-0h																															

Table 14-107. CFG_DATA_LL30_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL30_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.103 CFG_DATA_LL30_THRESHOLD Register (Offset = 1A0h) [reset = 0h]

CFG_DATA_LL30_THRESHOLD is shown in Figure 14-108 and described in Table 14-108.

Return to Summary Table.

Figure 14-108. CFG_DATA_LL30_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll30dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL30_WR_THRESHOLD							NU1	LL30_RD_THRESHOLD						
R-0h	R/W-0h							R-0h	R/W-0h						

Table 14-108. CFG_DATA_LL30_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll30dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL30_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL30_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.104 CFG_DATA_LL31 Register (Offset = 1A4h) [reset = 0h]

CFG_DATA_LL31 is shown in Figure 14-109 and described in Table 14-109.

Return to Summary Table.

Figure 14-109. CFG_DATA_LL31 Register

31	30	29	28	27	26	25	24
RESERVED			LL31_CRC_EN	LL31_LPHDR_EN	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED	LL31_SIZE						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
LL31_SIZE							LL31_FMT_IN
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
LL31_FMT_MAP	LL31_FMT		LL31_VCNUM		LL31_HS	LL31_HE	LL31_VALID
R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 14-109. CFG_DATA_LL31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	Reserved
28	LL31_CRC_EN	R/W	0h	0 : CRC is disbaled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL31_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data corresponind to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26-23	RESERVED	R/W	0h	Reserved
22-9	LL31_SIZE	R/W	0h	Configure the Size of the data in terms of the numer of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL31_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL31_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANEx_FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANEx_FMT_1_y
6-5	LL31_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL31_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL31_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL31_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL31_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

14.3.3.1.105 CFG_DATA_LL31_LPHDR_VAL Register (Offset = 1A8h) [reset = 0h]

CFG_DATA_LL31_LPHDR_VAL is shown in Figure 14-110 and described in Table 14-110.

Return to Summary Table.

Figure 14-110. CFG_DATA_LL31_LPHDR_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LL31_LPHDR_VAL																															
R/W-0h																															

Table 14-110. CFG_DATA_LL31_LPHDR_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LL31_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0xBBBBBBBB

14.3.3.1.106 CFG_DATA_LL31_THRESHOLD Register (Offset = 1ACh) [reset = 3F00h]

CFG_DATA_LL31_THRESHOLD is shown in Figure 14-111 and described in Table 14-111.

Return to Summary Table.

Figure 14-111. CFG_DATA_LL31_THRESHOLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3													ll31dman		
R-0h													R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2	LL31_WR_THRESHOLD							NU1	LL31_RD_THRESHOLD						
R-0h	R/W-3Fh							R-0h	R/W-0h						

Table 14-111. CFG_DATA_LL31_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll31dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA trasnfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	0h	
14-8	LL31_WR_THRESHOLD	R/W	3Fh	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	0h	
6-0	LL31_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

14.3.3.1.107 CFG_LVDS_MAPPING_LANE0_FMT_0 Register (Offset = 1B0h) [reset = 0h]

CFG_LVDS_MAPPING_LANE0_FMT_0 is shown in Figure 14-112 and described in Table 14-112.

Return to Summary Table.

Figure 14-112. CFG_LVDS_MAPPING_LANE0_FMT_0 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE0_FMT_0_H				CFG_LVDS_MAPPING_LANE0_FMT_0_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE0_FMT_0_F				CFG_LVDS_MAPPING_LANE0_FMT_0_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE0_FMT_0_D				CFG_LVDS_MAPPING_LANE0_FMT_0_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE0_FMT_0_B				CFG_LVDS_MAPPING_LANE0_FMT_0_A			
R/W-0h				R/W-0h			

Table 14-112. CFG_LVDS_MAPPING_LANE0_FMT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE0_FMT_0_H	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
27-24	CFG_LVDS_MAPPING_LANE0_FMT_0_G	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
23-20	CFG_LVDS_MAPPING_LANE0_FMT_0_F	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
19-16	CFG_LVDS_MAPPING_LANE0_FMT_0_E	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
15-12	CFG_LVDS_MAPPING_LANE0_FMT_0_D	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
11-8	CFG_LVDS_MAPPING_LANE0_FMT_0_C	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
7-4	CFG_LVDS_MAPPING_LANE0_FMT_0_B	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
3-0	CFG_LVDS_MAPPING_LANE0_FMT_0_A	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details

14.3.3.1.108 CFG_LVDS_MAPPING_LANE1_FMT_0 Register (Offset = 1B4h) [reset = 0h]

CFG_LVDS_MAPPING_LANE1_FMT_0 is shown in Figure 14-113 and described in Table 14-113.

Return to Summary Table.

Figure 14-113. CFG_LVDS_MAPPING_LANE1_FMT_0 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE1_FMT_0_H				CFG_LVDS_MAPPING_LANE1_FMT_0_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE1_FMT_0_F				CFG_LVDS_MAPPING_LANE1_FMT_0_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE1_FMT_0_D				CFG_LVDS_MAPPING_LANE1_FMT_0_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE1_FMT_0_B				CFG_LVDS_MAPPING_LANE1_FMT_0_A			
R/W-0h				R/W-0h			

Table 14-113. CFG_LVDS_MAPPING_LANE1_FMT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE1_FMT_0_H	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
27-24	CFG_LVDS_MAPPING_LANE1_FMT_0_G	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
23-20	CFG_LVDS_MAPPING_LANE1_FMT_0_F	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
19-16	CFG_LVDS_MAPPING_LANE1_FMT_0_E	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
15-12	CFG_LVDS_MAPPING_LANE1_FMT_0_D	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
11-8	CFG_LVDS_MAPPING_LANE1_FMT_0_C	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
7-4	CFG_LVDS_MAPPING_LANE1_FMT_0_B	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
3-0	CFG_LVDS_MAPPING_LANE1_FMT_0_A	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details

14.3.3.1.109 CFG_LVDS_MAPPING_LANE2_FMT_0 Register (Offset = 1B8h) [reset = 0h]

CFG_LVDS_MAPPING_LANE2_FMT_0 is shown in Figure 14-114 and described in Table 14-114.

Return to Summary Table.

Figure 14-114. CFG_LVDS_MAPPING_LANE2_FMT_0 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE2_FMT_0_H				CFG_LVDS_MAPPING_LANE2_FMT_0_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE2_FMT_0_F				CFG_LVDS_MAPPING_LANE2_FMT_0_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE2_FMT_0_D				CFG_LVDS_MAPPING_LANE2_FMT_0_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE2_FMT_0_B				CFG_LVDS_MAPPING_LANE2_FMT_0_A			
R/W-0h				R/W-0h			

Table 14-114. CFG_LVDS_MAPPING_LANE2_FMT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE2_FMT_0_H	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
27-24	CFG_LVDS_MAPPING_LANE2_FMT_0_G	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
23-20	CFG_LVDS_MAPPING_LANE2_FMT_0_F	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
19-16	CFG_LVDS_MAPPING_LANE2_FMT_0_E	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
15-12	CFG_LVDS_MAPPING_LANE2_FMT_0_D	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
11-8	CFG_LVDS_MAPPING_LANE2_FMT_0_C	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
7-4	CFG_LVDS_MAPPING_LANE2_FMT_0_B	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
3-0	CFG_LVDS_MAPPING_LANE2_FMT_0_A	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details

14.3.3.1.110 CFG_LVDS_MAPPING_LANE3_FMT_0 Register (Offset = 1BCh) [reset = 0h]

CFG_LVDS_MAPPING_LANE3_FMT_0 is shown in Figure 14-115 and described in Table 14-115.

Return to Summary Table.

Figure 14-115. CFG_LVDS_MAPPING_LANE3_FMT_0 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE3_FMT_0_H				CFG_LVDS_MAPPING_LANE3_FMT_0_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE3_FMT_0_F				CFG_LVDS_MAPPING_LANE3_FMT_0_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE3_FMT_0_D				CFG_LVDS_MAPPING_LANE3_FMT_0_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE3_FMT_0_B				CFG_LVDS_MAPPING_LANE3_FMT_0_A			
R/W-0h				R/W-0h			

Table 14-115. CFG_LVDS_MAPPING_LANE3_FMT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE3_FMT_0_H	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
27-24	CFG_LVDS_MAPPING_LANE3_FMT_0_G	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
23-20	CFG_LVDS_MAPPING_LANE3_FMT_0_F	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
19-16	CFG_LVDS_MAPPING_LANE3_FMT_0_E	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
15-12	CFG_LVDS_MAPPING_LANE3_FMT_0_D	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
11-8	CFG_LVDS_MAPPING_LANE3_FMT_0_C	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
7-4	CFG_LVDS_MAPPING_LANE3_FMT_0_B	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
3-0	CFG_LVDS_MAPPING_LANE3_FMT_0_A	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details

14.3.3.1.111 CFG_LVDS_MAPPING_LANE0_FMT_1 Register (Offset = 1C0h) [reset = 0h]

CFG_LVDS_MAPPING_LANE0_FMT_1 is shown in Figure 14-116 and described in Table 14-116.

Return to Summary Table.

Figure 14-116. CFG_LVDS_MAPPING_LANE0_FMT_1 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE0_FMT_1_H				CFG_LVDS_MAPPING_LANE0_FMT_1_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE0_FMT_1_F				CFG_LVDS_MAPPING_LANE0_FMT_1_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE0_FMT_1_D				CFG_LVDS_MAPPING_LANE0_FMT_1_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE0_FMT_1_B				CFG_LVDS_MAPPING_LANE0_FMT_1_A			
R/W-0h				R/W-0h			

Table 14-116. CFG_LVDS_MAPPING_LANE0_FMT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE0_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
27-24	CFG_LVDS_MAPPING_LANE0_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
23-20	CFG_LVDS_MAPPING_LANE0_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
19-16	CFG_LVDS_MAPPING_LANE0_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
15-12	CFG_LVDS_MAPPING_LANE0_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
11-8	CFG_LVDS_MAPPING_LANE0_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
7-4	CFG_LVDS_MAPPING_LANE0_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
3-0	CFG_LVDS_MAPPING_LANE0_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details

14.3.3.1.112 CFG_LVDS_MAPPING_LANE1_FMT_1 Register (Offset = 1C4h) [reset = 0h]

CFG_LVDS_MAPPING_LANE1_FMT_1 is shown in Figure 14-117 and described in Table 14-117.

Return to Summary Table.

Figure 14-117. CFG_LVDS_MAPPING_LANE1_FMT_1 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE1_FMT_1_H				CFG_LVDS_MAPPING_LANE1_FMT_1_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE1_FMT_1_F				CFG_LVDS_MAPPING_LANE1_FMT_1_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE1_FMT_1_D				CFG_LVDS_MAPPING_LANE1_FMT_1_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE1_FMT_1_B				CFG_LVDS_MAPPING_LANE1_FMT_1_A			
R/W-0h				R/W-0h			

Table 14-117. CFG_LVDS_MAPPING_LANE1_FMT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE1_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
27-24	CFG_LVDS_MAPPING_LANE1_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
23-20	CFG_LVDS_MAPPING_LANE1_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
19-16	CFG_LVDS_MAPPING_LANE1_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
15-12	CFG_LVDS_MAPPING_LANE1_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
11-8	CFG_LVDS_MAPPING_LANE1_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
7-4	CFG_LVDS_MAPPING_LANE1_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
3-0	CFG_LVDS_MAPPING_LANE1_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details

14.3.3.1.113 CFG_LVDS_MAPPING_LANE2_FMT_1 Register (Offset = 1C8h) [reset = 0h]

CFG_LVDS_MAPPING_LANE2_FMT_1 is shown in Figure 14-118 and described in Table 14-118.

Return to Summary Table.

Figure 14-118. CFG_LVDS_MAPPING_LANE2_FMT_1 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE2_FMT_1_H				CFG_LVDS_MAPPING_LANE2_FMT_1_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE2_FMT_1_F				CFG_LVDS_MAPPING_LANE2_FMT_1_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE2_FMT_1_D				CFG_LVDS_MAPPING_LANE2_FMT_1_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE2_FMT_1_B				CFG_LVDS_MAPPING_LANE2_FMT_1_A			
R/W-0h				R/W-0h			

Table 14-118. CFG_LVDS_MAPPING_LANE2_FMT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE2_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
27-24	CFG_LVDS_MAPPING_LANE2_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
23-20	CFG_LVDS_MAPPING_LANE2_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
19-16	CFG_LVDS_MAPPING_LANE2_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
15-12	CFG_LVDS_MAPPING_LANE2_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
11-8	CFG_LVDS_MAPPING_LANE2_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
7-4	CFG_LVDS_MAPPING_LANE2_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
3-0	CFG_LVDS_MAPPING_LANE2_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details

14.3.3.1.114 CFG_LVDS_MAPPING_LANE3_FMT_1 Register (Offset = 1CCh) [reset = 0h]

CFG_LVDS_MAPPING_LANE3_FMT_1 is shown in Figure 14-119 and described in Table 14-119.

Return to Summary Table.

Figure 14-119. CFG_LVDS_MAPPING_LANE3_FMT_1 Register

31	30	29	28	27	26	25	24
CFG_LVDS_MAPPING_LANE3_FMT_1_H				CFG_LVDS_MAPPING_LANE3_FMT_1_G			
R/W-0h				R/W-0h			
23	22	21	20	19	18	17	16
CFG_LVDS_MAPPING_LANE3_FMT_1_F				CFG_LVDS_MAPPING_LANE3_FMT_1_E			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CFG_LVDS_MAPPING_LANE3_FMT_1_D				CFG_LVDS_MAPPING_LANE3_FMT_1_C			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
CFG_LVDS_MAPPING_LANE3_FMT_1_B				CFG_LVDS_MAPPING_LANE3_FMT_1_A			
R/W-0h				R/W-0h			

Table 14-119. CFG_LVDS_MAPPING_LANE3_FMT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE3_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
27-24	CFG_LVDS_MAPPING_LANE3_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
23-20	CFG_LVDS_MAPPING_LANE3_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
19-16	CFG_LVDS_MAPPING_LANE3_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
15-12	CFG_LVDS_MAPPING_LANE3_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
11-8	CFG_LVDS_MAPPING_LANE3_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
7-4	CFG_LVDS_MAPPING_LANE3_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details
3-0	CFG_LVDS_MAPPING_LANE3_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for confiuration details

14.3.3.1.115 CFG_LVDS_GEN_0 Register (Offset = 1D0h) [reset = 04024C00h]

CFG_LVDS_GEN_0 is shown in Figure 14-120 and described in Table 14-120.

Return to Summary Table.

Figure 14-120. CFG_LVDS_GEN_0 Register

31	30	29	28	27	26	25	24
cpz		RESERVED	cbcrcen	cfdly			
R/W-0h		R/W-0h	R/W-0h	R/W-4h			
23	22	21	20	19	18	17	16
cmsbf	cpossel	RESERVED					
R/W-0h	R/W-0h	R/W-2h					
15	14	13	12	11	10	9	8
cclksel1	RESERVED			ccsmen	CFG_BIT_CLK_MODE	RESERVED	
R/W-0h	R/W-0h			R/W-1h	R/W-1h	R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				CFG_LVDS_LANE3_EN	CFG_LVDS_LANE2_EN	CFG_LVDS_LANE1_EN	CFG_LVDS_LANE0_EN
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-120. CFG_LVDS_GEN_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	cpz	R/W	0h	LVDS Clock config. 1 : Clock alignment enabled Others : Internal clock alignment not enabled This needs to be set to 0x1 for correct functionality
29	RESERVED	R/W	0h	Reserved
28	cbcrcen	R/W	0h	LVDS Frame CRC 0 : CRC is not sent at the end of LVDS Frame 1 : CRC is sent at the end of the LVDS Frame
27-24	cfdly	R/W	4h	LVDS FIFO Initial Threshold. This is a Static configuration and sould be set to a fixed value as mention in the Programming model
23	cmsbf	R/W	0h	1 : Data is sent out on the LVDS lane MSB first 0 : Data is sent out on the LVDS lane LSB first
22	cpossel	R/W	0h	0 : When a new chirp is starting, align first sample start to negedge of DDR clock. 1 : When a new chirp is starting, align first sample start to posedge of DDR clock (recommended)
21-16	RESERVED	R/W	2h	Reserved
15	cclksel1	R/W	0h	0 : DDR mode clock mux 1 : SDR mode clock mux
14-12	RESERVED	R/W	0h	Reserved
11	ccsmen	R/W	1h	0 : Regular operation 1 : Continuous Streaming Mode Enabled
10	CFG_BIT_CLK_MODE	R/W	1h	Bit Clock Mode 0 : SDR clocking mode 1 : DDR clocking mode
9-4	RESERVED	R/W	0h	Reserved
3	CFG_LVDS_LANE3_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 3 is disbaled 1 : LVDS Lane 3 is enabled
2	CFG_LVDS_LANE2_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 2 is disbaled 1 : LVDS Lane 2 is enabled
1	CFG_LVDS_LANE1_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 1 is disbaled 1 : LVDS Lane 1 is enabled
0	CFG_LVDS_LANE0_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 0 is disbaled 1 : LVDS Lane 0 is enabled

14.3.3.1.116 CFG_LVDS_GEN_1 Register (Offset = 1D4h) [reset = 50h]

CFG_LVDS_GEN_1 is shown in Figure 14-121 and described in Table 14-121.

Return to Summary Table.

Figure 14-121. CFG_LVDS_GEN_1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					c3c3l	RESERVED	
R-0h					R/W-0h	R-0h	

Table 14-121. CFG_LVDS_GEN_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	c3c3l	R/W	0h	LVDS Only Programming: 0 : Regular Operation 1 : Enable 3Ch-3Lane mode in LVDS. Refer to Programming model for more details
1-0	RESERVED	R/W	0h	Reserved

14.3.3.1.117 CFG_LVDS_GEN_2 Register (Offset = 1D8h) [reset = 0h]

CFG_LVDS_GEN_2 is shown in Figure 14-122 and described in Table 14-122.

Return to Summary Table.

Figure 14-122. CFG_LVDS_GEN_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_LVDS_GEN_2																															
R/W-0h																															

Table 14-122. CFG_LVDS_GEN_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_LVDS_GEN_2	R/W	0h	CFG_LVDS_GEN_2[0]: Configure LSB/MSB first for CRC. This feature is supported only when the field CFG_LVDS_GEN_0[28] is set to 1 0 -> The calculated value of 32-bit Ethernet polynomial CRC is swapped and sent out, clear this bit if data is set to LSB first (CFG_LVDS_GEN_0[23]=0) but CRC should be MSB first or vice-versa 1 -> The calculated value of 32-bit Ethernet polynomial CRC is sent out without swapping, set this bit if both data and CRC should have same format (LSB/MSB first) CFG_LVDS_GEN_2[1]: Configure value of frame clock during inter frame period 0 -> Frame clock is held low 1 -> Frame clock is held high CFG_LVDS_GEN_2[2]: Configure frame clock period. This feature is supported only when the field CFG_LVDS_GEN_0[28] is set to 1 0 -> 32-bit CRC is transmitted as single packet with frame clock set to 16h16l (16 high 16 low) configuration 1 -> 32-bit CRC is transmitted as two packets with frame clock set to 8h8l (8 high 8 low) configuration for each packet CFG_LVDS_GEN_2[3]: Configure bit clock during inter frame period 0 -> Bit clock toggles during inter frame period 1 -> Bit clock does not toggle during inter frame period, the value of bit clock is held low. This feature is supported when DDR clock is selected (CFG_LVDS_GEN_0[10]=1) and first data sample is driven on posedge of DDR clock (CFG_LVDS_GEN_0[22]=1) CFG_LVDS_GEN_2[4]: Configure CRC inversion. This feature is supported only when the field CFG_LVDS_GEN_0[28] is set to 1 0 -> The calculated value of 32-bit Ethernet polynomial CRC is inverted and sent out 1 -> The calculated value of 32-bit Ethernet polynomial CRC is sent out without inversion CFG_LVDS_GEN_2[5]: Enable/disable the calibration mode, in this mode frame clock will follow data lane[0] 0 -> Calibration mode is disabled 1 -> Calibration mode is enabled

14.3.3.1.118 CFG_MASK_REG0 Register (Offset = 1DCh) [reset = FFFFFFFFh]

CFG_MASK_REG0 is shown in Figure 14-123 and described in Table 14-123.

Return to Summary Table.

Figure 14-123. CFG_MASK_REG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_MASK_REG0																															
R/W-FFFFFFFFh																															

Table 14-123. CFG_MASK_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG0	R/W	FFFFFFFFh	Mask Register field corresponding to STAT_CBUFF_REG0. Refer STAT_CBUFF_REG0 for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occuruence 1 : Event is masked. No interrupt will be generated on occurrence

14.3.3.1.119 CFG_MASK_REG1 Register (Offset = 1E0h) [reset = FFFFFFFFh]

CFG_MASK_REG1 is shown in Figure 14-124 and described in Table 14-124.

Return to Summary Table.

Figure 14-124. CFG_MASK_REG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_MASK_REG1																															
R/W-FFFFFFFFh																															

Table 14-124. CFG_MASK_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG1	R/W	FFFFFFFFh	Mask Register field corresponding to STAT_CBUFF_REG1. Refer STAT_CBUFF_REG1 for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occuruence 1 : Event is masked. No interrupt will be generated on occurrence

14.3.3.1.120 CFG_MASK_REG2 Register (Offset = 1E4h) [reset = FFFFFFFFh]

CFG_MASK_REG2 is shown in Figure 14-125 and described in Table 14-125.

Return to Summary Table.

Figure 14-125. CFG_MASK_REG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_MASK_REG2																															
R/W-FFFFFFFFh																															

Table 14-125. CFG_MASK_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG2	R/W	FFFFFFFFh	Mask Register field corresponding to STAT_LVDS_REG0. Refer STAT_LVDS_REG0 for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occuruence 1 : Event is masked. No interrupt will be generated on occurrence

14.3.3.1.121 CFG_MASK_REG3 Register (Offset = 1E8h) [reset = FFFFFFFFh]

CFG_MASK_REG3 is shown in Figure 14-126 and described in Table 14-126.

Return to Summary Table.

Figure 14-126. CFG_MASK_REG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_MASK_REG3																															
R/W-FFFFFFFFh																															

Table 14-126. CFG_MASK_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG3	R/W	FFFFFFFFh	RESERVED

14.3.3.1.122 STAT_CBUFF_REG0 Register (Offset = 1ECh) [reset = 0h]

STAT_CBUFF_REG0 is shown in Figure 14-127 and described in Table 14-127.

Return to Summary Table.

Figure 14-127. STAT_CBUFF_REG0 Register

31	30	29	28	27	26	25	24
STAT_CBUFF_REG0_OTHERS							
R-0h							
23	22	21	20	19	18	17	16
STAT_CBUFF_REG0_OTHERS							
R-0h							
15	14	13	12	11	10	9	8
STAT_CBUFF_REG0_OTHERS			S_FRAME_DONE	S_CHIRP_DONE	RESERVED		
R-0h			R-0h	R-0h	R-0h		
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 14-127. STAT_CBUFF_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	STAT_CBUFF_REG0_OTHERS	R	0h	Reserved for future enhancement
12	S_FRAME_DONE	R	0h	Indicates that CBUFF has completed sending out data for the current Frame
11	S_CHIRP_DONE	R	0h	Indicates that CBUFF has completed sending out data for the current Chirp
10-0	RESERVED	R	0h	Reserved

14.3.3.1.123 STAT_CBUFF_REG1 Register (Offset = 1F0h) [reset = 800h]

STAT_CBUFF_REG1 is shown in Figure 14-128 and described in Table 14-128.

Return to Summary Table.

Figure 14-128. STAT_CBUFF_REG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						S_FRAME_ERR	S_CHIRP_ERR
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 14-128. STAT_CBUFF_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	S_FRAME_ERR	R	0h	Indicates the FrameStart arrived before CBUFF has completed sending out data for all the Chirps programmed
16	S_CHIRP_ERR	R	0h	Indicates tha the chirpAvailable from ADCBuffer arrived before CBUFF has completed sending out the previous Chirp data.
15-10	RESERVED	R	0h	Reserved

14.3.3.1.124 STAT_CBUFF_REG2 Register (Offset = 1F4h) [reset = 0h]

STAT_CBUFF_REG2 is shown in Figure 14-129 and described in Table 14-129.

Return to Summary Table.

Figure 14-129. STAT_CBUFF_REG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STAT_CBUFF_REG2																															
R-0h																															

Table 14-129. STAT_CBUFF_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STAT_CBUFF_REG2	R	0h	RESERVED. This does not have coresponding clear or mask

14.3.3.1.125 STAT_CBUFF_REG3 Register (Offset = 1F8h) [reset = 0h]

STAT_CBUFF_REG3 is shown in Figure 14-130 and described in Table 14-130.

Return to Summary Table.

Figure 14-130. STAT_CBUFF_REG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STAT_CBUFF_REG3																															
R-0h																															

Table 14-130. STAT_CBUFF_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STAT_CBUFF_REG3	R	0h	RESERVED. This does not have coresponding clear or mask

14.3.3.1.126 CLR_CBUFF_REG0 Register (Offset = 20Ch) [reset = 0h]

CLR_CBUFF_REG0 is shown in Figure 14-131 and described in Table 14-131.

Return to Summary Table.

Figure 14-131. CLR_CBUFF_REG0 Register

31	30	29	28	27	26	25	24
RESERVED							
0h							
23	22	21	20	19	18	17	16
RESERVED							
0h							
15	14	13	12	11	10	9	8
RESERVED			C_FRAME_DONE	C_CHIRP_DONE	RESERVED		
0h			0h	0h	0h		
7	6	5	4	3	2	1	0
RESERVED							
0h							

Table 14-131. CLR_CBUFF_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED		0h	Reserved
12	C_FRAME_DONE		0h	Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field
11	C_CHIRP_DONE		0h	Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field
10-0	RESERVED		0h	Reserved

14.3.3.1.127 STAT_CBUFF_ECC_REG Register (Offset = 21Ch) [reset = 0h]

STAT_CBUFF_ECC_REG is shown in Figure 14-132 and described in Table 14-132.

Return to Summary Table.

Figure 14-132. STAT_CBUFF_ECC_REG Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2							
R-0h							
15	14	13	12	11	10	9	8
NU2						seccdbe	seccsbe
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
NU1		seccadd					
R-0h		R-0h					

Table 14-132. STAT_CBUFF_ECC_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU2	R	0h	
9	seccdbe	R	0h	0 : No Double bit error 1 : Indicates a double bit error has occurred
8	seccsbe	R	0h	0 : No Single bit error 1 : Indicates a single bit error has occurred
7-6	NU1	R	0h	
5-0	seccadd	R	0h	6-bit address where the ECC error occurred. It is valid when either seccsbe or seccdbe is set. If none of them is set, then the addr does not mean anything.

14.3.3.1.128 MASK_CBUFF_ECC_REG Register (Offset = 220h) [reset = 300h]

MASK_CBUFF_ECC_REG is shown in Figure 14-133 and described in Table 14-133.

Return to Summary Table.

Figure 14-133. MASK_CBUFF_ECC_REG Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2							
R-0h							
15	14	13	12	11	10	9	8
NU2						meccdbe	meccsbe
R-0h						R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
NU1							
R-0h							

Table 14-133. MASK_CBUFF_ECC_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU2	R	0h	
9	meccdbe	R/W	1h	0 : Double bit error indications are unmasked 1 : Double bit error indications are Masked
8	meccsbe	R/W	1h	0 : Single bit error indications are unmasked 1 : Single bit error indications are Masked
7-0	NU1	R	0h	

14.3.3.1.129 CLR_CBUFF_ECC_REG Register (Offset = 224h) [reset = 0h]

CLR_CBUFF_ECC_REG is shown in Figure 14-134 and described in Table 14-134.

Return to Summary Table.

Figure 14-134. CLR_CBUFF_ECC_REG Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2							
R-0h							
15	14	13	12	11	10	9	8
NU2						ceccdbe	ceccsbe
R-0h						0h	0h
7	6	5	4	3	2	1	0
NU1							ceccadd
R-0h							0h

Table 14-134. CLR_CBUFF_ECC_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU2	R	0h	
9	ceccdbe		0h	Clear Register field corresponding to STAT_CBUFF_ECC. Write 0x1 to Clear the field
8	ceccsbe		0h	Clear Register field corresponding to STAT_CBUFF_ECC. Write 0x1 to Clear the field
7-1	NU1	R	0h	
0	ceccadd		0h	Clear Register field corresponding to STAT_CBUFF_ECC. Write 0x1 to Clear the field

14.3.3.1.130 STAT_SAFETY Register (Offset = 228h) [reset = 0h]

STAT_SAFETY is shown in Figure 14-135 and described in Table 14-135.

Return to Summary Table.

Figure 14-135. STAT_SAFETY Register

31	30	29	28	27	26	25	24
SAF_UNUSED1							
R-0h							
23	22	21	20	19	18	17	16
SAF_UNUSED1							
R-0h							
15	14	13	12	11	10	9	8
SAF_UNUSED1							SAF_CHIRP_ERR
R-0h							R-0h
7	6	5	4	3	2	1	0
SAF_CRC							
R-0h							

Table 14-135. STAT_SAFETY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	SAF_UNUSED1	R	0h	RESERVED
8	SAF_CHIRP_ERR	R	0h	Safety Error. Indicates tha the chirpAvailable from ADCBuffer arrived before CBUFF has completed sending out the previous Chirp data.
7-0	SAF_CRC	R	0h	Indicates a CRC error between ADCBuffer and CBUFF. 0 : No Error Non Zero : Error

14.3.3.1.131 MASK_SAFETY Register (Offset = 22Ch) [reset = FFFFFFFFh]

MASK_SAFETY is shown in Figure 14-136 and described in Table 14-136.

Return to Summary Table.

Figure 14-136. MASK_SAFETY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK_SAFETY																															
R/W-FFFFFFFFh																															

Table 14-136. MASK_SAFETY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASK_SAFETY	R/W	FFFFFFFFh	Mask Register field corresponding to STAT_SAFETY. Refer STAT_SAFETY for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occuruence 1 : Event is masked. No interrupt will be generated on occurrence

14.3.3.1.132 CLR_SAFETY Register (Offset = 230h) [reset = 0h]

CLR_SAFETY is shown in Figure 14-137 and described in Table 14-137.

Return to Summary Table.

Figure 14-137. CLR_SAFETY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR_SAFETY																															
0h																															

Table 14-137. CLR_SAFETY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CLR_SAFETY		0h	Clear Register field corresponding to STAT_SAFETY. Write 0x1 to Clear the field

14.4 CSI2

14.4.1 CSI2 Overview

The CSI-2 Specification defines standard data transmission and control interfaces between transmitter and receiver. The data transmission interface (referred as CSI-2) is a unidirectional differential serial interface with data and clock signals; the physical layer of this interface is the MIPI Alliance Specification for D-PHY.

The physical layer for a CSI-2 implementation is composed of four unidirectional data lanes and one clock lane.

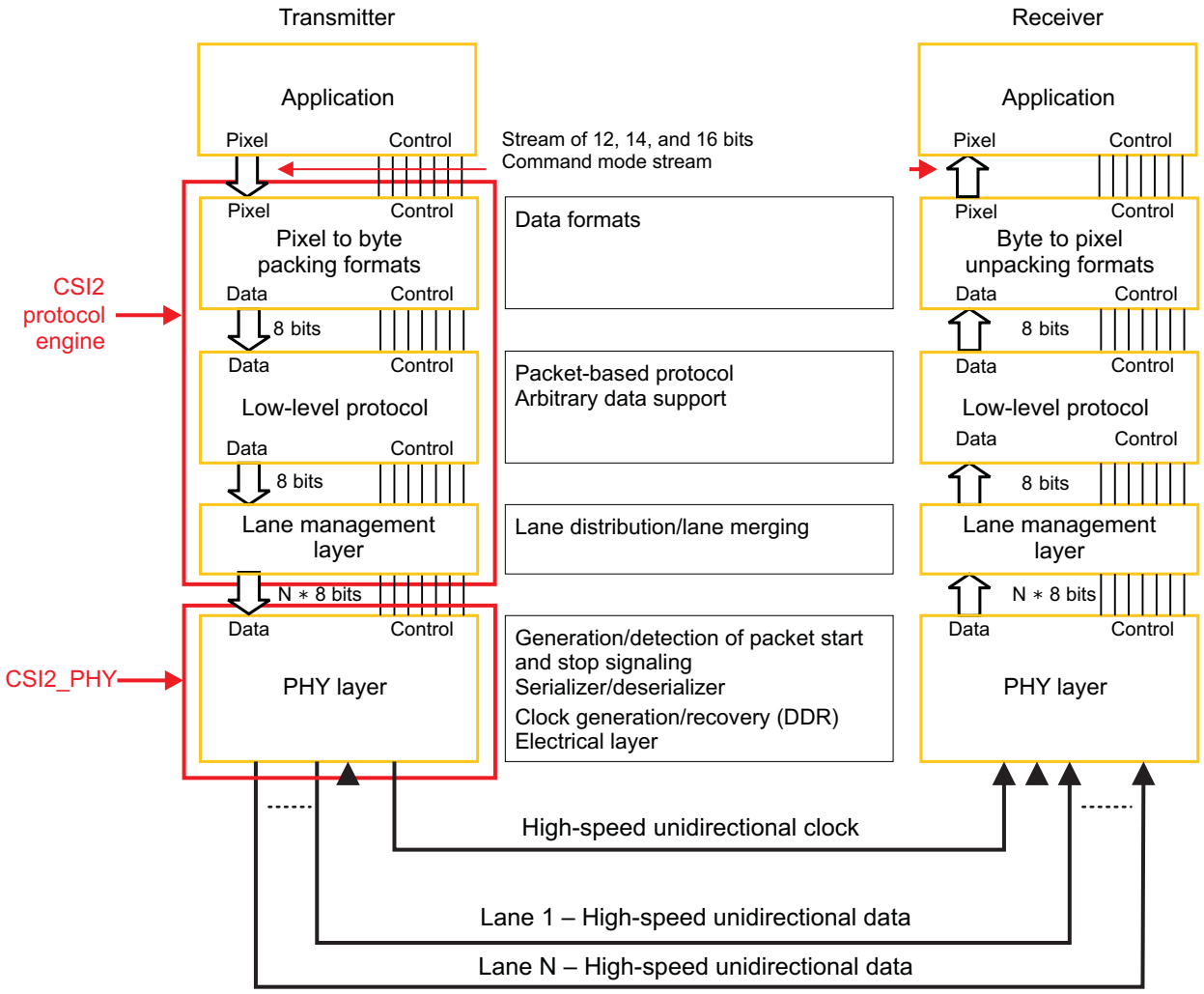
Main features:

- Supports MIPI-CSI2 (up to four data lanes and one clock lane)
- Supports command mode only
- RAW8, RAW12, and RAW14 formats are supported for command mode
- Supports up to four data-configurable lanes, in addition to the clock signaling
- Maximum data rate of 600 Mbps per data pair for four-data lane configuration, and 900 Mbps for up to three-data lane configuration
- Data splitter for 2-data, 3-data, and 4-data lane configurations
- Connection to the CSI2_PHY complex I/O through PPI
- Error-correction code (ECC) and checksum generation

14.4.1.1 CSI2 Environment

Figure 14-138 shows the high-level data flow of the CSI2 transmitter/receiver.

Figure 14-138. CSI2 Transmitter/Receiver Data Flow



14.4.1.1.1 CSI2 PHY

Table 14-138 lists the CSI2 I/Os.

Table 14-138. I/O Description of CSI2

Signal Name		I/O[1]	Description	Value at Reset
CSI2_dx0 CSI2_dy0	lane 1	I/O	Serial data/clock lane	N/A
CSI2_dx1 CSI2_dy1	lane 2	I/O	Serial data/clock lane	N/A
CSI2_dx2 CSI2_dy2	lane 3	I/O	Serial data/clock lane	N/A
CSI2_dx3 CSI2_dy3	lane 4	I/O	Serial data/clock lane	N/A
CSI2_dx4 CSI2_dy4	lane 5	I/O	Serial data lane only	N/A

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

Lanes support four operating modes:

- HS mode: High-speed transmit mode
- LP mode: Low-power transmit mode (also called low-power state [LPS])
- Off mode: Lane is off.

NOTE: The DSS_REG:CSI_CFG1 register in the DSS_REG of the device provides the following control over the CSI2 lanes:

- Enable and disable each lane
- Pulldown enable and disable on each lane

14.4.1.1.1.1 Data/Clock Configuration

Data-clock signaling consists of one to four data pairs and one clock pair. The minimum configuration is one data pair and one clock pair.

- The data signal carries the bit serial data. The CSI2 transmitter in the host sends the data in-quadrature with the dual data rate (DDR) clock in HS mode. The data is transmitted byte-wise LSB first.
- The clock signal carries the DDR clock signal in HS transmission.
- Software users must configure the order of the data lanes to indicate the byte order while splitting the byte stream for each CSI2_PHY into bytes.

Table 14-139 lists some of the CSI2 lane configurations.

NOTE: All combinations for the order of the clock and data lanes are supported (except clock cannot be on lane 5) but not explicitly described in Table 14-139.

Table 14-139. CSI2 Lane Configuration

CSI2_PHY Lane Configuration	Data/Clock Lane Position					Description
	1	2	3	4	5	
Mode CLK + DATA1						Single data lane
	CLK	DATA1				
	DATA1	CLK				
Mode CLK + DATA1 + DATA2						Two data lanes
	CLK	DATA1	DATA2			
	CLK	DATA2	DATA1			
	DATA1	CLK	DATA2			
	DATA2	CLK	DATA1			
	DATA1	DATA2	CLK			
	DATA2	DATA1	CLK			
Mode CLK + DATA1 + DATA2 + DATA3						Three data lanes
	CLK	DATA1	DATA2	DATA3		
	CLK	DATA3	DATA2	DATA1		
	DATA1	CLK	DATA2	DATA3		
	DATA1	CLK	DATA3	DATA2		
	DATA2	CLK	DATA1	DATA3		
	DATA2	CLK	DATA3	DATA1		
	DATA3	CLK	DATA1	DATA2		
	DATA3	CLK	DATA2	DATA1		
	DATA3	DATA1	CLK	DATA2		

Table 14-139. CSI2 Lane Configuration (continued)

CSI2_PHY Lane Configuration	Data/Clock Lane Position					Description
	1	2	3	4	5	
	DATA3	DATA2	CLK	DATA1		
	DATA3	DATA2	CLK	DATA1		
	DATA1	DATA2	CLK	DATA3		
	DATA2	DATA3	CLK	DATA1		
	DATA1	DATA3	CLK	DATA2		
	DATA1	DATA2	DATA3	CLK		
	DATA3	DATA2	DATA1	CLK		
Mode CLK + DATA1 + DATA2 + DATA3 + DATA4						Four data lanes
	CLK	DATA1	DATA2	DATA3	DATA4	
	DATA3	DATA1	CLK	DATA2	DATA4	
	DATA4	DATA2	CLK	DATA1	DATA3	
	DATA2	DATA1	CLK	DATA3	DATA4	
	DATA4	DATA3	CLK	DATA1	DATA2	
	DATA3	DATA4	CLK	DATA2	DATA1	
	DATA1	DATA2	CLK	DATA4	DATA3	
	DATA1	DATA4	CLK	DATA3	DATA2	
	DATA1	DATA3	CLK	DATA4	DATA2	

14.4.1.1.2 Low Level Protocol

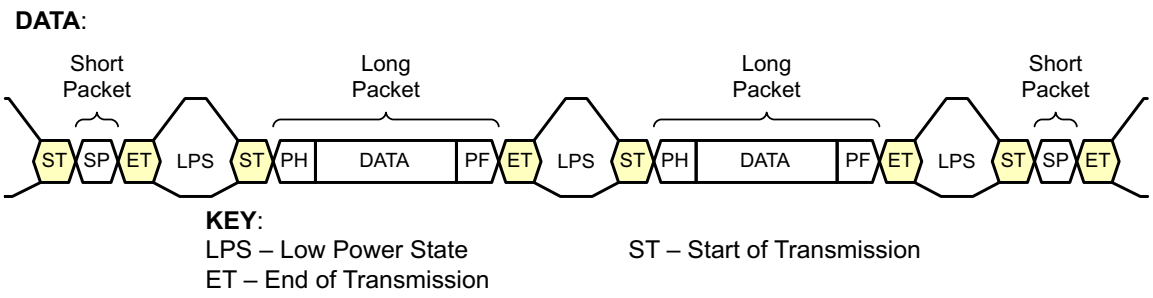
The Low Level Protocol (LLP) is a byte orientated, packet-based protocol that supports the transport of arbitrary data using short and long packet formats. For simplicity, all examples in this section are single lane configurations.

Low level protocol features:

- Transport of arbitrary data (payload independent)
- 8-bit word size
- Support for up to four interleaved VCs on the same link
- Special packets for frame start, frame end, line start, and line end information
- Descriptor for the type, pixel depth, and format of application-specific payload data
- ECC for 1-bit or 2-bit error detection in the header
- 16-bit checksum code for error detection

Figure 14-139 shows the protocol layer with short and long packets.

Figure 14-139. Low Level Protocol Packet Overview



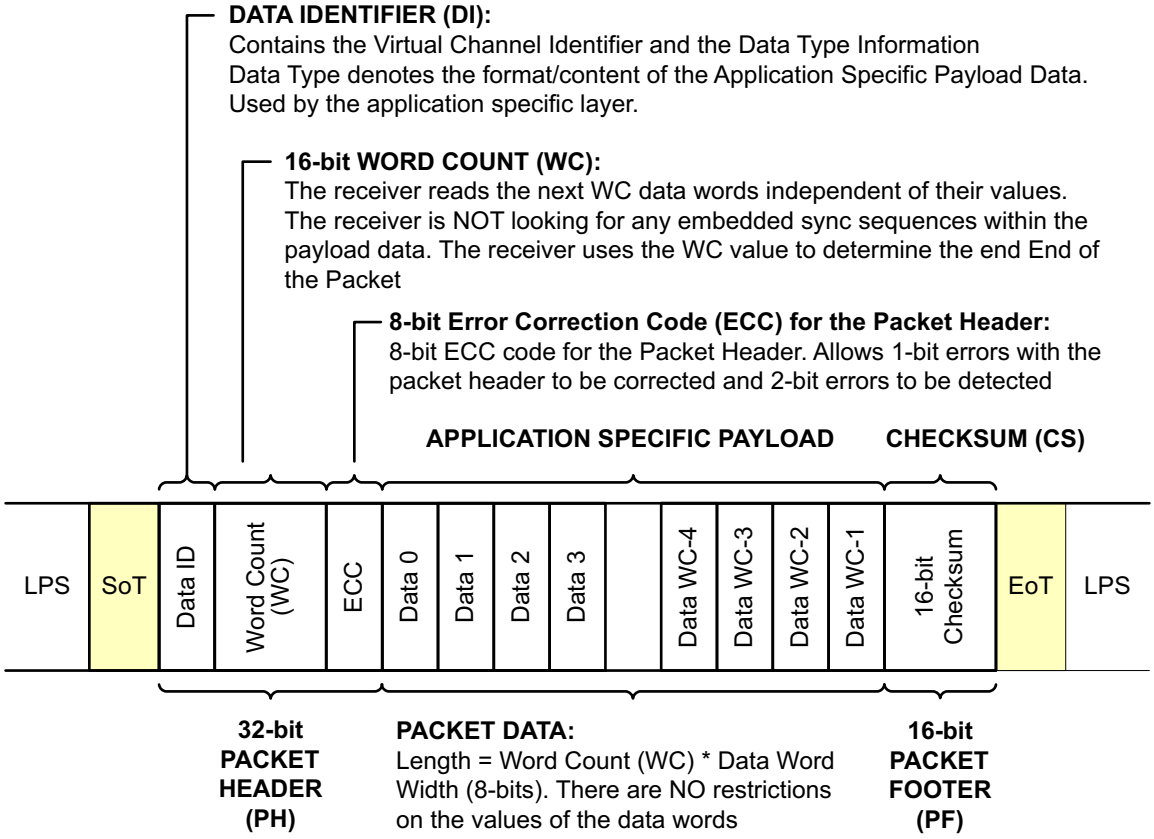
14.4.1.1.2.1 Low-Level Protocol Packet Format

Two packet structures are defined for low-level protocol communication: long packets and short packets. For each packet structure exit from the low-power state, the start of transmission (SoT) sequence indicates the start of the packet. The end of transmission (EoT) sequence, followed by the low-power state, indicates the end of the packet.

14.4.1.1.2.1.1 Low-Level Protocol Long Packet Format

Figure 14-140 shows the structure of the low level protocol long packet. A long packet is identified by data types 0x10 to 0x37. See Table 14-140 for a description of the data types. A long packet consists of three elements: a 32-bit packet header (PH), an application-specific data payload with a variable number of 8-bit data words, and a 16-bit packet footer (PF). The packet header is further composed of three elements: an 8-bit data identifier, a 16-bit word count field, and an 8-bit ECC. The packet footer has one element, a 16-bit checksum. See Section 14.4.1.1.2.2 through Section 14.4.1.1.2.3.2 for further descriptions of the packet elements.

Figure 14-140. Long Packet Structure



The data identifier defines the virtual channel for the data and the data type for the application-specific payload data.

The word count defines the number of 8-bit data words in the data payload between the end of the packet header and the start of the packet footer. Neither the packet header nor the packet footer shall be included in the word count.

The error correction code (ECC) byte allows single-bit errors to be corrected, and 2-bit errors to be detected in the packet header. This includes both the data identifier value and the word count value.

After the end of the packet header, the receiver reads the next word count 8-bit data words of the data payload. While reading the data payload, the receiver does not look for any embedded sync codes. Therefore, there are no limitations on the value of a data word.

When the receiver has read the data payload, it reads the checksum in the packet footer. In the generic case, the length of the data payload is a multiple of 8-bit data words. In addition, each data format may impose additional restrictions on the length of the payload data, such as a multiple of four bytes.

Each byte is transmitted least significant bit first. Payload data may be transmitted in any byte order, restricted only by data format requirements. Multi-byte elements such as word count, checksum, and the short packet 16-bit data field are transmitted least significant byte first.

After the EoT sequence, the receiver begins looking for the next SoT sequence.

14.4.1.1.2.2 Short Packet

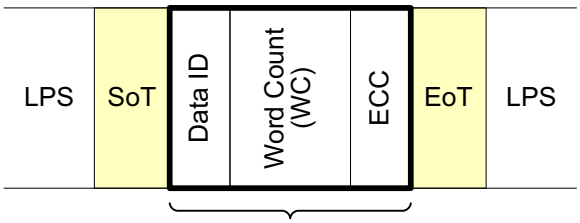
Figure 14-141 shows the structure of the low level protocol short packet. A short packet is identified by data types 0x00 to 0x0F. See Table 14-140 for a description of the data types. A short packet contains only a packet header; a packet footer is not present. The word count field in the packet header is replaced by a short packet data field.

For frame synchronization data types, the short packet data field is the frame number. For line synchronization data types, the short packet data field is the line number.

For generic short packet data types, the content of the short packet data field is user-defined.

The error correction code (ECC) byte allows single-bit errors to be corrected, and 2-bit errors to be detected in the short packet.

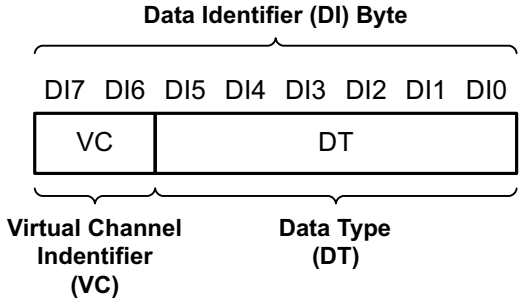
Figure 14-141. Short Packet Structure



14.4.1.1.2.3 Data Identifier (DI)

The data identifier byte contains the virtual channel identifier (VC) value and the data type (DT) value, as shown in Figure 14-142. The virtual channel identifier is contained in the 2 MS bits of the data identifier byte The data type value is contained in the six LS bits of the data identifier byte.

Figure 14-142. Data Identifier Byte



14.4.1.1.2.3.1 Virtual Channel Identifier

The purpose of the virtual channel identifier is to provide separate channels for different data flows interleaved in the data stream.

The virtual channel identifier number is in the top two bits of the data identifier byte. The receiver monitors the virtual channel identifier and de-multiplexes the interleaved video streams to their appropriate channel. A maximum of four data streams is supported; valid channel identifiers are 0 to 3. The virtual channel identifiers in the peripherals should be programmable, to allow the host processor to control how the data streams are de-multiplexed. The principle of logical channels is presented in Figure 14-143.

Figure 14-143. Logical Channel Block Diagram (Receiver)

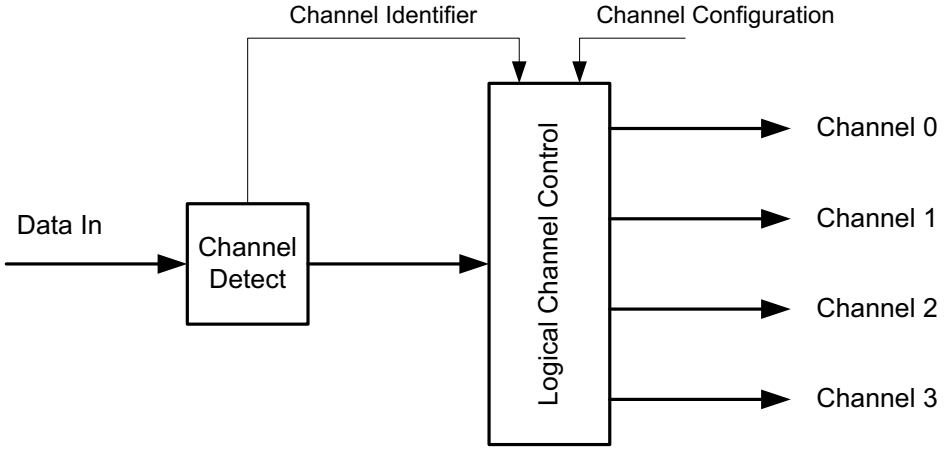
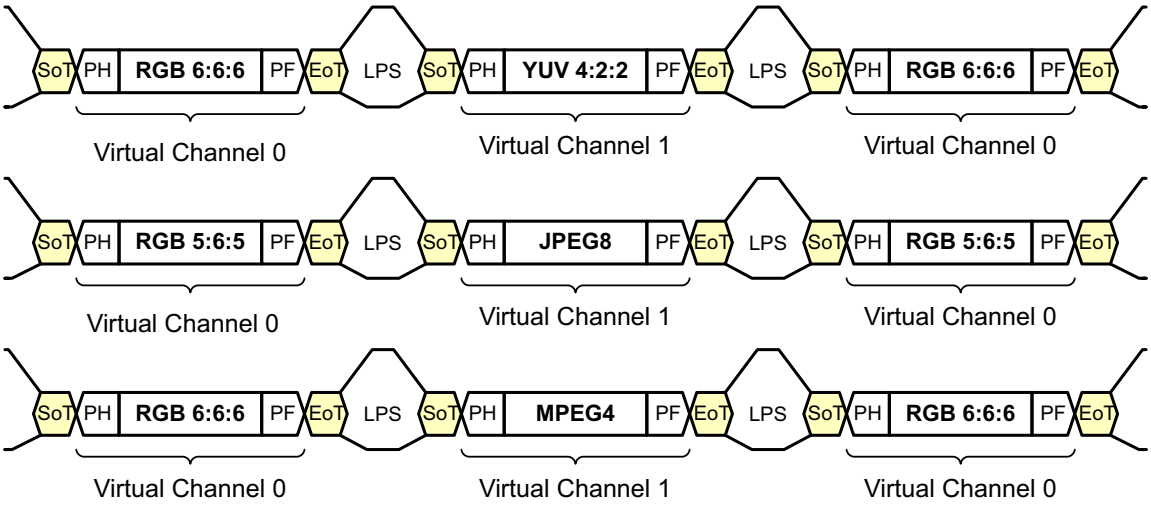


Figure 14-144 illustrates an example of data streams utilizing virtual channel support.

Figure 14-144. Interleaved Video Data Streams Examples



KEY:
LPS – Low Power State
SoT – Start of Transmission
EoT – End of Transmission
PH – Packet Header
PF – Packet Footer

14.4.1.1.2.3.2 Data Type (DT)

The data type value specifies the format and content of the payload data. A maximum of sixty-four data types are supported.

There are eight different data type classes, as shown in Table 14-140. Within each class, there are up to eight different data type definitions. The first two classes denote short packet data types. The remaining six classes denote long packet data types.

Table 14-140. Data Type Classes

Data Type	Description
0x00 to 0x07	Synchronization short packet data types
0x08 to 0x0F	Generic short packet data types
0x10 to 0x17	Generic long packet data types
0x18 to 0x1F	YUV data
0x20 to 0x27	RGB data
0x28 to 0x2F	RAW data
0x30 to 0x37	User-defined byte-based data
0x38 to 0x3F	Reserved

14.4.1.1.3 CSI2 Multilane Layer

A layer consists of lane splitter logic to split the incoming byte stream into a serial stream. The bits are sent with the LSB first. The number of active lanes is configurable through a register. The order of the lanes is configurable. The number of lanes can be changed only in ULPS or when all data lanes are in STOP state.

14.4.1.1.3.1 SoT and EoT in Multilane Configurations

Because a HS transmission is composed of an arbitrary number of bytes that may not be an integer multiple of the number of lanes, some lanes may run out of data before others. Therefore, the lane management layer, as it buffers up the final set of less-than-N bytes, deasserts its valid data signal into lanes for which there is no further data. Although all lanes start simultaneously with parallel start-of-transmissions (SoTs), each lane operates independently and may complete the HS transmission before the other lanes, sending an end-of-transmission (EoT) one cycle (byte) earlier.

14.4.1.1.3.2 Lane Splitter

The lane splitter can split the byte stream into two, three, or four lanes (for one lane, the splitter is bypassed). Figure 14-145 through Figure 14-148 show the byte position in each serial link for one-, two-, three-, and four-data lane configurations. The byte stream always starts from lane 1. It finishes on one of the lanes, depending on the number of bytes to send and the number of lanes. The splitter module is used only when packets are sent using HS mode. In LS mode, only data lane 1 is used.

Figure 14-145. Four-Data Lane Configuration

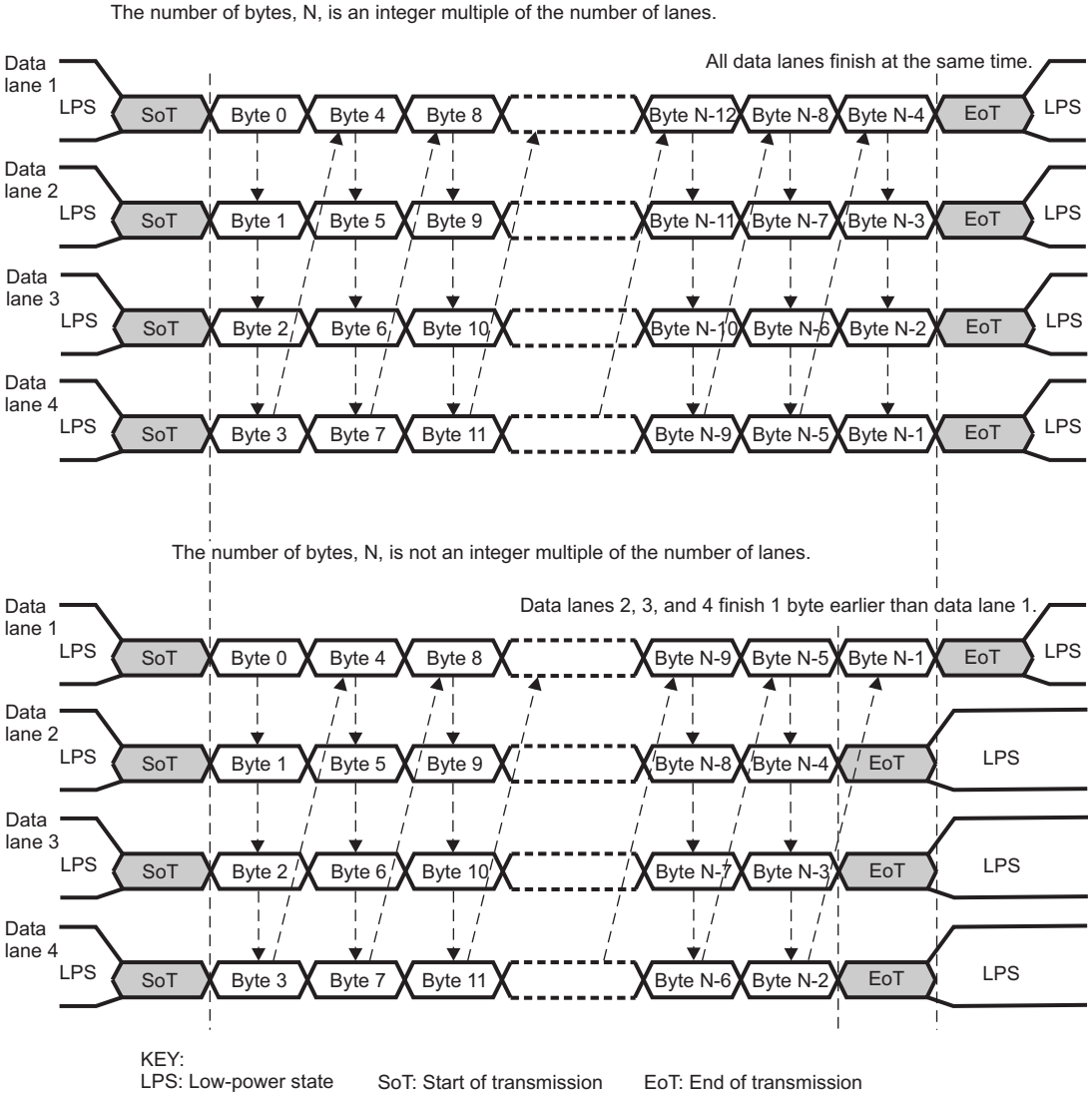


Figure 14-146. Three-Data Lane Configuration

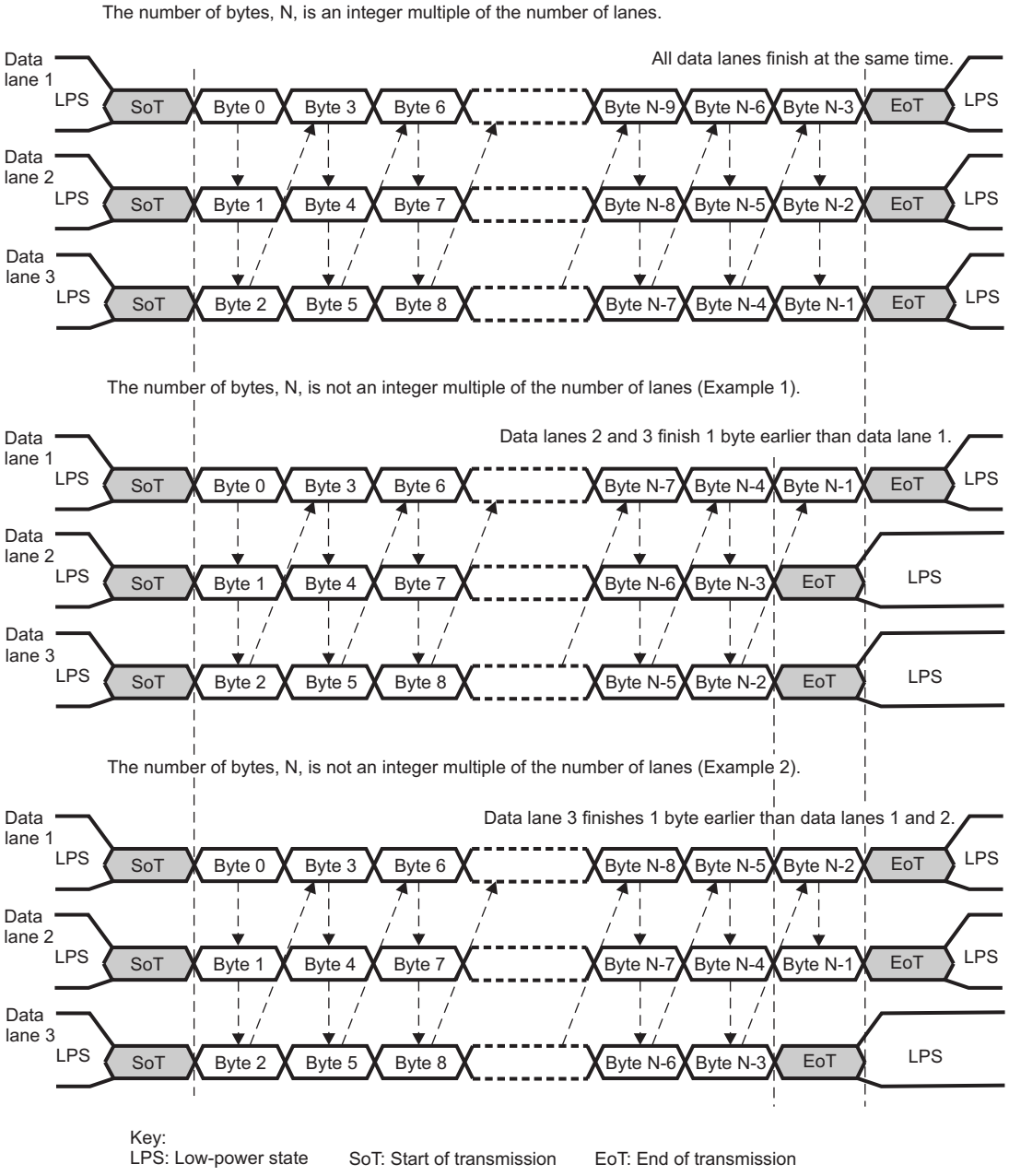


Figure 14-147. Two-Data Lane Configuration

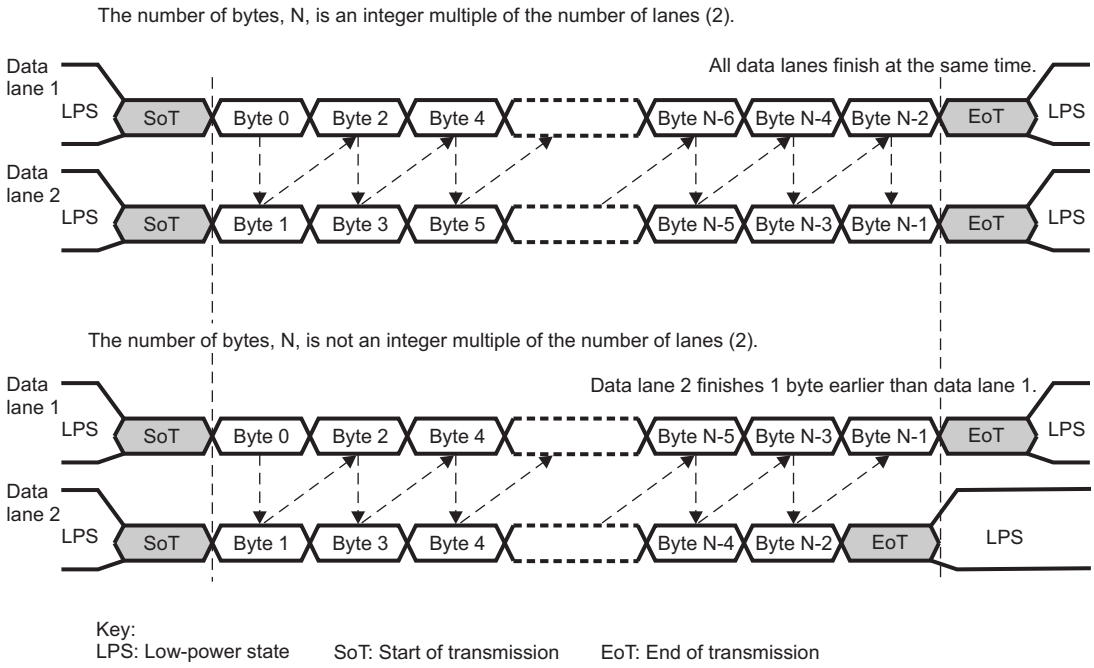
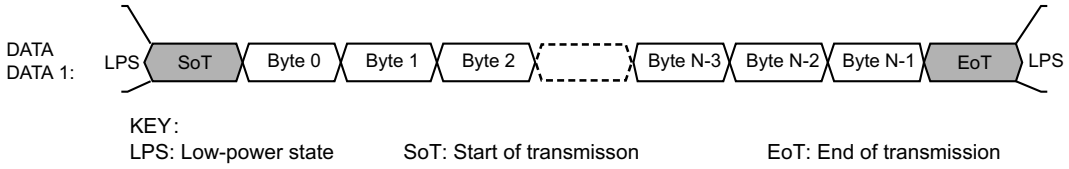


Figure 14-148. One-Data Lane Configuration



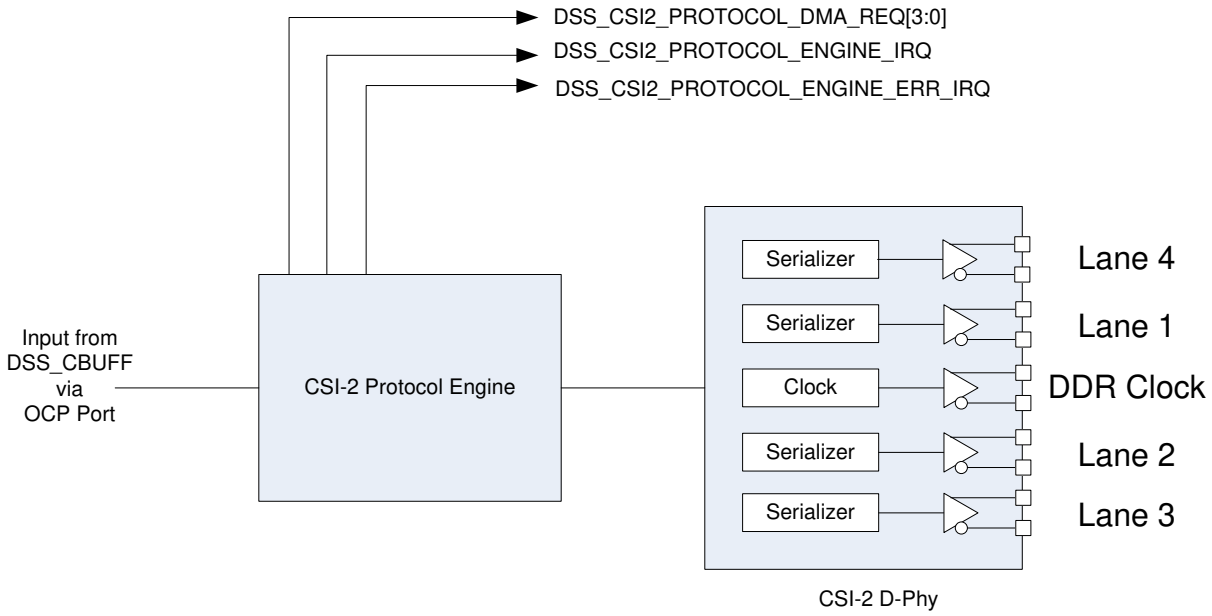
14.4.2 CSI2 Protocol Engine and Phy

This section describes the functions of the CSI2 module.

14.4.2.1 CSI2 Protocol Overview

The CSI2 protocol engine receives data from the common buffer, generates the ECC and checksum, and splits the data into byte streams to the CSI2_PHY to be sent using HS protocol.

Figure 14-149. CSI2 Protocol Engine



14.4.2.2 CSI2 Global Register Controls

Before receiving data from the CSI2 complex I/O, the CSI2_PHY SCP registers in the CSI2 complex I/O must be configured. Table 14-141 lists the register access width limitations for all the CSI2 modules.

Table 14-141. Register Access Width Limitations

Register Name	Register Access Width
All CSI2 complex I/O registers (CSI2_PHY SCP)	32-bit only
CSI2_VC_LONG_PACKET_HEADER_i	32-bit only
CSI2_VC_SHORT_PACKET_HEADER_i	32-bit only
CSI2_VC_LONG_PACKET_PAYLOAD_i	16- and 32-bit
All other CSI2 protocol engine registers	8-, 16-, and 32-bit

The CSI2 protocol engine is globally controlled by the CSI2_CTRL register. The interface to the complex I/O is enabled by setting the CSI2_CTRL[0] IF_EN bit. When the interface is disabled, it is possible to provide data to the TX FIFO and read pending data in the RX FIFO. When the CSI2_CTRL[0] IF_EN bit is set to 1, the pending packets are sent to the CSI2 complex I/O.

14.4.2.3 CSI2 Transfer Modes

14.4.2.3.1 Command Mode

The CSI2_VC_LONG_PACKET_HEADER registers are used for the header of long packets; the CSI2_VC_SHORT_PACKET_HEADER registers are used for the short packets.

For the payload, the CSI2_VC_LONG_PACKET_PAYLOAD register is used. Each 32-bit payload data is written to the CSI2_VC_LONG_PACKET_PAYLOAD register from the CBUFF.

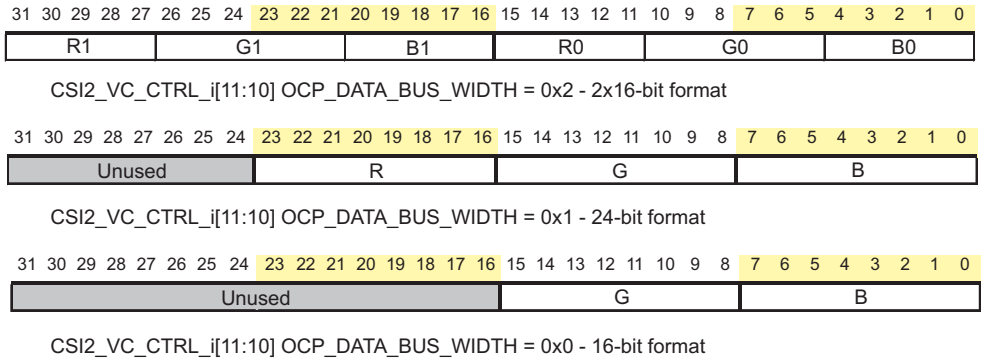
The CSI2_VC_CTRL_i[11:10] OCP_DATA_BUS_WIDTH bit field is used to define the size of the data on the slave port for the write access:

- 0x0: 16-bit data width (LSB of the 32-bit interface port data bus)
- 0x1: 24-bit data width (LSB of the 32-bit interface port data bus)
- 0x2: 2x16-bit data width (first pixel on the LSB of the 32-bit Interface port data bus and second pixel on the MSB of the 32-bit interface port data bus for the same interface access)
- 0x3: 32-bit data width (no bit, no byte, no pixel manipulation)

The device only uses the mode with 0x3.

Figure 14-150 shows the different data width configurations of the L3 interconnect slave port.

Figure 14-150. Interface Data Configuration



The write into the CSI2_VC_LONG_PACKET_HEADER_i register is required before accessing the CSI2_VC_LONG_PACKET_PAYLOAD_i register. The CSI2 protocol engine must extract the length of the payload and discard the extra data sent using the CSI2_VC_LONG_PACKET_PAYLOAD_i register. Writes into the CSI2_VC_LONG_PACKET_HEADER_i register are valid only if the VC is enabled; otherwise, the write is ignored.

The device always uses the VC_0 FIFO and the registers in the protocol engine. VC management is within the CBUFF.

14.4.2.4 Clock Requirements

The serial clock generated by the CSI2 host must be a continuous clock. It is software-programmed through the CSI2_CLK_CTRL[13] DDR_CLK_ALWAYS_ON bit: This bit can be programmed only when the interface is disabled (the CSI2_CTRL[0] IF_EN bit set to 0).

The CSI2_CLK_CTRL[20] LP_CLK_ENABLE bit is used to enable or disable the clock. When disabled, the value of the CSI2_CLK_CTRL[12:0] LP_CLK_DIVISOR bit field is ignored and does not have to be programmed by software users.

14.4.2.4.1 Timing Parameters for an LP to HS Transaction

Figure 14-151 shows the timing requirement when switching the data and clock lane state from LP to HS.

lists the LP to HS timing parameters.

Figure 14-151. LP to HS Timing

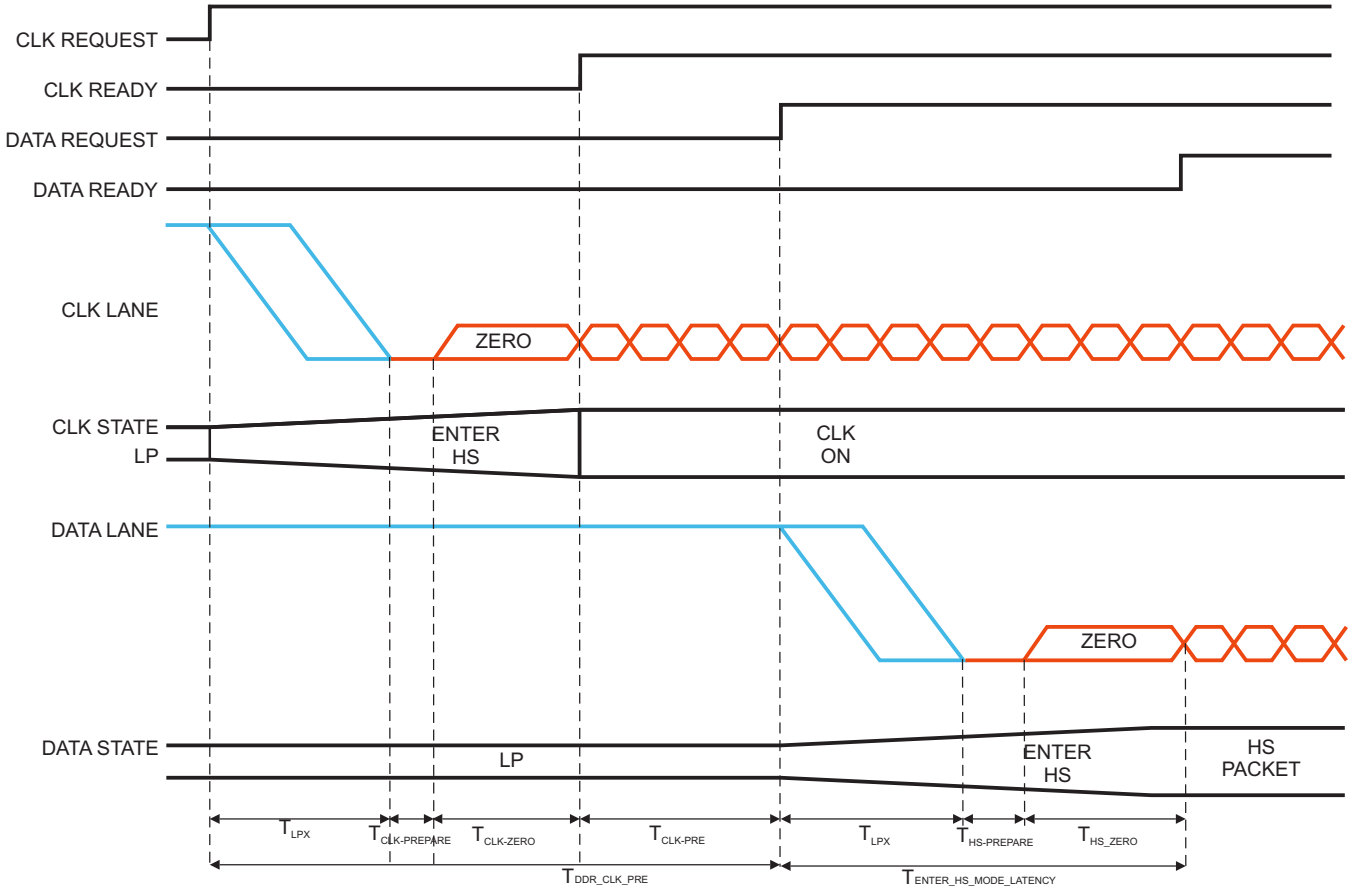


Table 14-142. LP to HS Timing Parameters

Timings	Description	Registers/Associated Bit Fields	Register Settings	Timing Seen on the Line
T_{LPX}	Length of any low-power state period	CSI2_PHY_REGISTER1[20:16] REG_TLPXBY2 The value set in this bit field is half of the T_{LPX}	CEIL (25 ns / DDR_Clock_Period)	CEIL (2 * REG_TLPXBY2/4) * 4 * DDR_Clock_Period
$T_{CLK-PREPARE}$	Time to drive the CLK lane to LP-00 state, to prepare for HS clock transmission	CSI2_PHY_REGISTER2[7:0] REG_TCLKPREPARE	CEIL (65 ns / DDR_Clock_Period)	REG_TCLKPREPARE * DDR_Clock_Period + (~– 25 ns – +5 ns)
$T_{CLK-ZERO}$	Time to drive the CLK lane to HS-0 state, before starting the clock	CSI2_PHY_REGISTER1[7:0] REG_TCLKZERO	CEIL (265 ns / DDR_Clock_Period)	{CEIL [(REG_TCLKZERO + 3)/4] * 4 + CEIL (REG_TCLKPREPARE/4) * 4 – REG_TCLKPREPARE + 2} * DDR_Clock_Period + (~ 0 ns – +5 ns)
$T_{CLK-PRE}$	Time that the HS clock must be driven before any associated data lane begins the transition from LP to HS mode	N/A	N/A	DDR_CLK_PRE - T_{LPX} - $T_{CLK-PREPARE}$ - $T_{CLK-ZERO}$
$T_{HS-PREPARE}$	Time to drive the data lane to LP-00 state, to prepare for HS packet transmission	CSI2_PHY_REGISTER0[31:24] REG_THSPREPARE	CEIL (70 ns / DDR_Clock_Period) + 2	REG_THSPREPARE * DDR_Clock_Period + (–26.5 ns – + 4 ns)
$T_{HS-ZERO}$ + $T_{HS-PREPARE}$	$T_{HS-ZERO}$: Time to drive the data lane to HS-0 state before the synchronous sequence.	CSI2_PHY_REGISTER0[23:16] REG_THSPRPR_THSZERO	CEIL (175 ns / DDR_Clock_Period) + 2	{CEIL [(N + 3)/4] * 4 + CEIL (M/4) * 4 + 3} * DDR_Clock_Period + (~ – 29 ns – 0 ns). Where N = REG_THSPREPARE_THSZERO – REG_THSPREPARE M = REG_THSPREPARE
$T_{DDR\_CLK\_PRE}$	Time between the CLK lane request assertion and the data request assertion to switch the data lanes to HS	CSI2_CLK_TIMING[15:8] DDR_CLK_PRE	CEIL [(T_{LPX} + $T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$ + $T_{CLK-PRE}$) / $T_{TXBYTECLKHS}$] (1)(2)	
$T_{ENTER\_HS\_MODE\_LATENCY}$ (3)	Time to enter data lane into HS mode.	CSI2_VM_TIMING7[31:16] ENTER_HS_MODE_LATENCY	CEIL [(T_{LPX} + $T_{HS-PREPARE}$ + $T_{HS-ZERO}$) / $T_{TXBYTECLKHS}$] (1)	

(1) The timings seen on the line should be used to determine the register value.
(2) See the MIPI D-PHY specification for the $T_{CLK-PRE}$ value.
(3) ENTER_HS_MODE_LATENCY timing applies only to video mode. It does not need to be programmed in command mode.

In the example in Table 14-143, the DDR clock = 400 MHz; TxByteClkHS = 100 MHz.

Table 14-143. LP to HS Timing Parameters Example for 400-MHz DDR Clock

Timings	Registers/Associated Bit Fields	Default Register Settings (Programmed at Reset)	Timing Seen on the Line
T_{LPX}	CSI2_PHY_REGISTER1[20:16] REG_TLPXBY2	10	50 ns
$T_{CLK-PREPARE}$	CSI2_PHY_REGISTER2[7:0] REG_TCLKPREPARE	26	40–70 ns
$T_{CLK-ZERO}$	CSI2_PHY_REGISTER1[7:0] REG_TCLKZERO	106	280–285 ns

Table 14-143. LP to HS Timing Parameters Example for 400-MHz DDR Clock (continued)

Timings	Registers/Associated Bit Fields	Default Register Settings (Programmed at Reset)	Timing Seen on the Line
$T_{CLK-PRE}$ (1)	N/A	N/A	80 ns
$T_{HS-PREPARE}$	CSI2_PHY_REGISTER0[31:24] REG_THSPREPARE	30	48.5–79 ns
$T_{HS-ZERO}$	CSI2_PHY_REGISTER0[23:16] REG_THSPRPR_THSZERO	72	178.5–207.5 ns
$T_{DDR\_CLK\_PRE}$	CSI2_CLK_TIMING[15:8] DDR_CLK_PRE	45–49 (2)	450–490 ns
$T_{ENTER\_HS\_MODE\_LATENCY}$	CSI2_VM_TIMING7[31:16] ENTER_HS_MODE_LATENCY	24 * $T_{TXBYTECLKHS}$ or 96 * DDR_CLOCK 34 * $T_{TXBYTECLKHS}$ or 136 * DDR_CLOCK (2)	277–336.5 ns

(1) See the MIPI D-PHY specification for the $T_{CLK-PRE}$ value.
(2) Register setting calculated based on the values in the Timing Seen on the Line column.

14.4.2.4.2 Timing Parameters for an HS to LP Transaction

Figure 14-152 shows the timing requirement when switching the state of the data and clock lanes from HS to LP. lists the HS to LP timing parameters.

Figure 14-152. HS to LP Timing

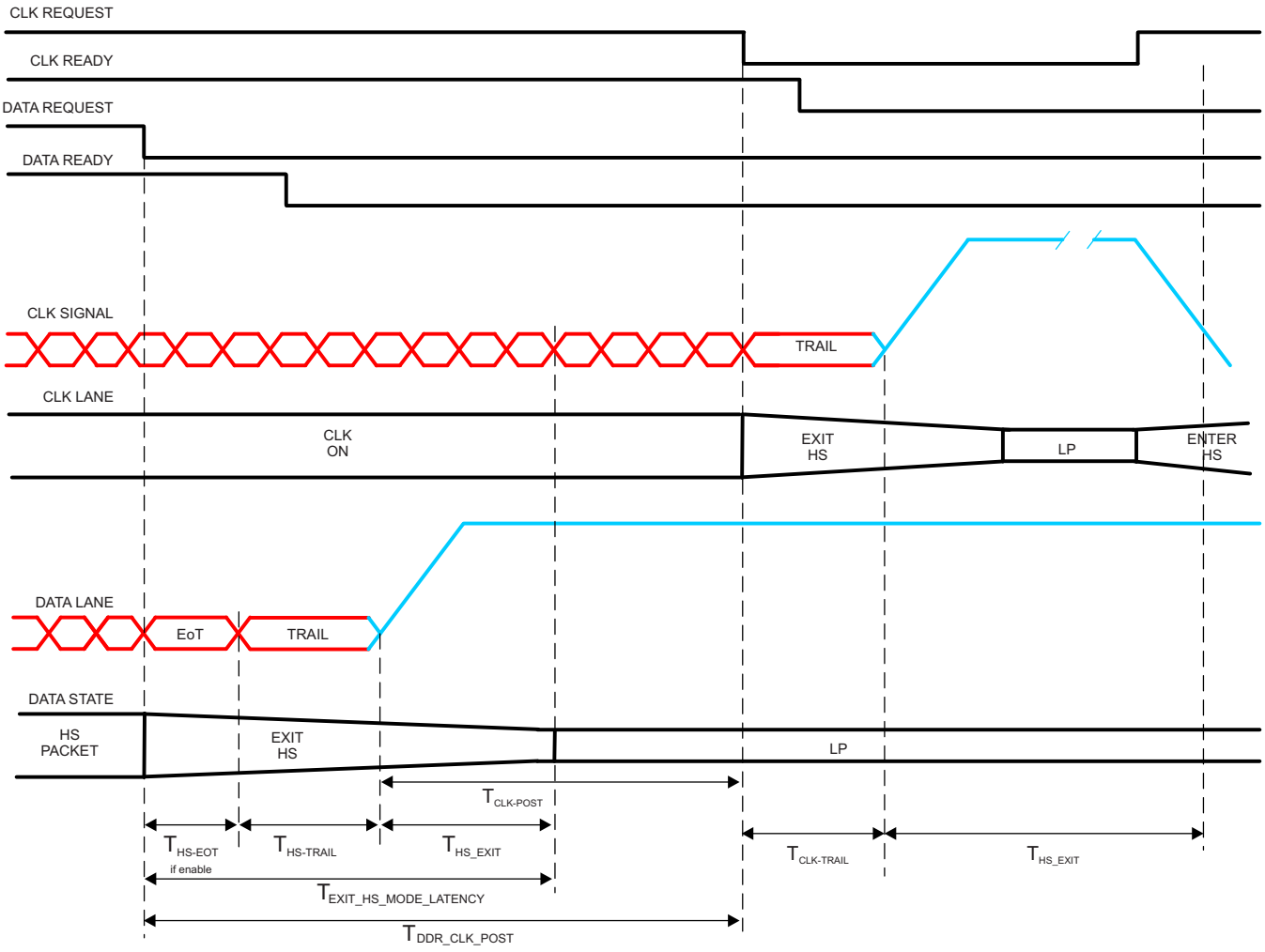


Table 14-144. HS to LP Timing Parameters

Timings	Description	Registers/Associated Bit Fields	Register Settings	Timing Seen on the Line
T_{HS-EOT}	If EoT is enabled, a delay is added to EXIT_HS_MODE_LATENCY to send the EoT packet. The EoT period depends on the number of data lanes.	N/A	N/A	DIVROUNDUP (4, NB_DATA_LANES) Thus: One data lane = Four DDR clocks Two data lanes = Two DDR clocks
$T_{HS-TRAIL}$	Time to drive flipped differential state after last payload data bit of an HS transmission burst	CSI2_PHY_REGISTER0[15:8] REG_THSTRAIL	CEIL (60 ns / DDR_Clock_Period) + 5	{CEIL [(REG_THSTRAIL + 3)/4] * 4 – 2.75} * DDR_Clock_Period + (~ 0 ns – 5 ns)
$T_{HS-EXIT}$	Time to drive data lane to LP-11 state, after HS burst	CSI2_PHY_REGISTER0[7:0] REG_THSEXIT	CEIL (145 ns / DDR_Clock_Period)	[CEIL (REG_THSEXIT/4) * 4] * DDR_Clock_Period – (~ 3 ns – 45 ns)
$T_{CLK-POST}$	Time that the transmitter must continue sending HS clock after the last associated data lane has transitioned to LP mode	N/A	N/A	DDR_CLK_POST - T_{HS-EOT} [(1)] - $T_{HS-TRAIL}$
$T_{CLK-TRAIL}$	Time to drive HS differential state after last payload clock bit of an HS transmission burst	CSI2_PHY_REGISTER1[15:8] REG_TCLKTRAIL	CEIL (60 ns / DDR_Clock_Period) + 2	{CEIL [(REG_TCLKTRAIL + 3)/4] * 4 – 1.5} * DDR_Clock_Period + (~ 0 ns – 5 ns)
$T_{DDR\_CLK\_POST}$	Time between the data lane request deassertion and the CLK request deassertion to switch the data lanes into LP mode. The DDR_CLK_POST value must follow the rule: $T_{DDR\_CLK\_POST} \geq T_{HS-TRAIL} + T_{HS-EOT} + T_{CLK-POST}$	CSI2_CLK_TIMING[7:0] DDR_CLK_POST	CEIL [($T_{HS-TRAIL}$ + T_{HS-EOT} [(1)] + $T_{CLK-POST}$) / $T_{TXBYTECLKHS}$] [(2)] [(3)]	
$T_{EXIT\_HS\_MODE\_LATENCY}$ [(4)]	Time to exit HS mode	CSI2_VM_TIMING7[15:0] EXIT_HS_MODE_LATENCY	CEIL [($T_{HS-TRAIL}$ + $T_{HS-EXIT}$ + T_{HS-EOT} [(1)]) / $T_{TXBYTECLKHS}$] [(2)]	

(1) If T_{HS-EOT} is enabled
(2) The timings seen on the line should be used to determine the register value.
(3) See the MIPI D-PHY specification for the$T_{CLK-POST}$ value.
(4) EXIT_HS_MODE_LATENCY timing applies only to video mode. It does not need to be programmed in command mode.

In the example in Table 14-145, the DDR clock = 400 MHz; TxByteClkHS = 100 MHz; two data lanes.

Table 14-145. HS to LP Timing Parameters Example for 400-MHz DDR Clock and Two Data Lanes

Timings	Registers/Associated Bit Fields	Default Register Settings (Programmed at Reset)	Timing Seen on the Line (T_{HS-EOT} Enabled)	Timing Seen on the Line (T_{HS-EOT} Disabled)
T_{HS-EOT}	N/A	N/A	5 ns	N/A
$T_{HS-TRAIL}$	CSI2_PHY_REGISTER0 [15:8] REG_THSTRAIL	29	73.125–78.125 ns	73.125–78.125 ns
$T_{HS-EXIT}$	CSI2_PHY_REGISTER0 [7:0] REG_THSEXIT	58	105–147 ns	105–147 ns
$T_{CLK-POST}$ [(1)]	N/A	N/A	585 ns	580 ns

(1) See the MIPI D-PHY specification.

Table 14-145. HS to LP Timing Parameters Example for 400-MHz DDR Clock and Two Data Lanes (continued)

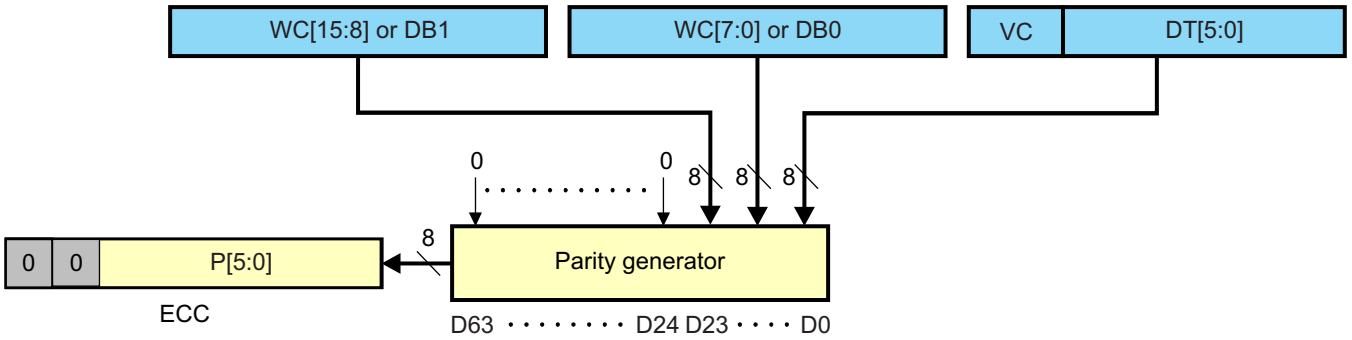
Timings	Registers/Associated Bit Fields	Default Register Settings (Programmed at Reset)	Timing Seen on the Line (T_{HS-EOT} Enabled)	Timing Seen on the Line (T_{HS-EOT} Disabled)
$T_{CLK-TRAIL}$	CSI2_PHY_REGISTER1 [15:8] REG_TCLKTRAIL	26	76.25–81.25 ns	76.25–81.25 ns
DDR_CLK_POST	CSI2_CLK_TIMING[7:0] DDR_CLK_POST	66 * TxByteClkHS or 264 * DDR_CLOCK	658.125–663.125 ns	653.125–658.125 ns
		67 * TxByteClkHS or 268 * DDR_CLOCK (2)		
EXIT_HS_MODE_LATENCY	CSI2_VM_TIMING7[15:0] EXIT_HS_MODE_LATENCY	19 * TxByteClkHS or 76 * DDR_CLOCK	183.125–230.125 ns	178.125–225.125 ns
		24 * TxByteClkHS or 96 * DDR_CLOCK (2)		

(2) Register setting calculated based on the values in Timing Seen on the Line (T_{HS-EOT} Enabled) column.

14.4.2.5 ECC Generation

The CSI2 protocol uses a 4-byte PH. Because ECC generation requires a fixed word length of 64-bits, the packet headers must be padded with additional bits to form a full 8-byte value for ECC generation and checking. The PH minus the ECC byte should occupy bits D[23:0], and the pad bits should occupy bits D[63:24]. All padding bits must be 0 to generate the ECC byte. ECC can be generated using a parallel approach, as shown in Figure 14-153.

Figure 14-153. 64-Bit ECC Generation on TX Side



The ECC generation/check can be enabled and disabled by software. It is defined by a common bit for all the VCs:

- The CSI2_VC_CTRL_i[8] ECC_TX_EN bit enables and disables ECC generation.

The ECC can be provided while writing the ECC value directly into the CSI2_VC_LONG_PACKET_HEADER_i and CSI2_VC_SHORT_PACKET_HEADER_i registers. The CSI2_VC_CTRL_i[8] ECC_TX_EN bit indicates whether the ECC value will be calculated or whether the value written in the register will be used instead.

14.4.2.6 Checksum Generation for Long Packet Payloads

Long packets are comprised of a PH protected by an ECC byte and a payload of 0 to 2^{16} - 1 bytes. To detect the errors during the transmission of long packets, a checksum is calculated over the payload portion of the data packet. For the special case of a 0-length payload, the 2-byte checksum is set to 0xFFFF. The checksum can only indicate the presence of one or more errors in the payload. Unlike ECC, the checksum does not enable error correction. For this reason, checksum calculation is not useful for some unidirectional CSI2 implementations, because the peripheral has no way to report errors to the host processor. Checksum generation and transmission is mandatory for host processors sending long packets

to peripherals; it is optional for peripherals transmitting long packets to the host processor. However, the format of long packets is fixed; the peripherals that do not support checksum generation must transmit 2 bytes with a value of 0x0000 in place of the checksum bytes when sending long packets to the host processor. The host processor must disable checksum checking for received long packets from peripherals that do not support checksum generation.

Checksum must be realized as a 16-bit CRC with a generator polynomial of $x^{16}+x^{12}+x^5+x^0$. The LSByte is sent first, followed by the MSByte. Within the byte, the LSB is sent first.

Figure 14-154 shows the checksum transmission.

Figure 14-154. Checksum Transmission

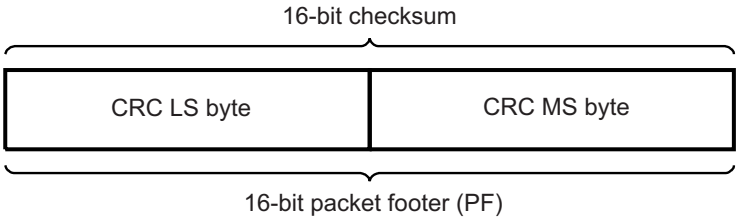
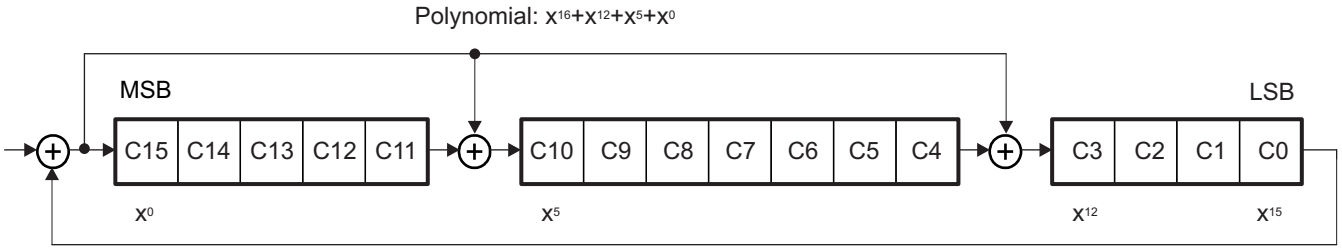


Figure 14-155 shows the CRC implementation. The CRC shift register is initialized to 0xFFFF before packet data enters. Packet data not including the PH then enters as a bitwise data stream from the left, LSB first. Each bit is fed through the CRC shift register before it is passed to the output for transmission to the peripheral. After all bytes in the packet payload have passed through the CRC shift register, the shift register contains the checksum. C15 contains the MSB of the checksum, and C0 contains the LSB of the 16-bit checksum. The checksum is then appended to the data stream and sent to the receiver.

Figure 14-155. 16 Bit CRC Generation Using a Shift Register



Checksum generation/check can be enabled and disabled by software. It is defined by a common bit for all the VCs:

- The CSI2_CTRL[1] CS_RX_EN bit enables and disables checksum generation.
- The CSI2_VC_CTRL_i[7] CS_TX_EN bit enables and disables checksum generation.

14.4.2.7 Software Reset

The CSI2 protocol engine can be reset by software. This reset can be done for debug purposes or after a protocol error and has the same effect as the hardware reset. The CSI2 protocol engine can be reset by setting the CSI2_SYSCONFIG[1] SOFT_RESET bit to 1. Software can monitor the CSI2_SYSSTATUS[0] RESET_DONE status bit to wait for the reset to complete. If after five reads the CSI2_SYSSTATUS[0] RESET_DONE status bit still returns 0, it can be assumed that an error occurred during the reset stage.

NOTE: This software reset is optional because a hardware reset is always performed on the CSI2 protocol engine at device reset.

14.4.2.8 Power Management

The power-management behavior of the CSI2 protocol engine is controlled by the CSI2_SYSCONFIG register. This register controls the way the module interacts with the PRCM module. The CSI2_SYSCONFIG[0] AUTO_IDLE bit must be set to 1 (default value) to enable automatic clock gating in the module.

The CSI2 protocol engine implements a handshake protocol on the interconnect slave port with the PRCM module. The protocol engine provides control signal CIO_CLK_ICG to gate the SCPClk (MSS_VCLK/4). It allows reduction of the power consumption of the complex I/O while the CSI2 link is not in use (CSI2_CLK_CTRL[14] CIO_CLK_ICG).

14.4.2.9 Power Control of CSI2 PHY Complex I/O

The CSI2 protocol engine can control and send power commands for the CSI2 complex I/O module.

14.4.2.9.1 Complex I/O Power Control Commands

14.4.2.9.1.1 Complex I/O Power Control Commands

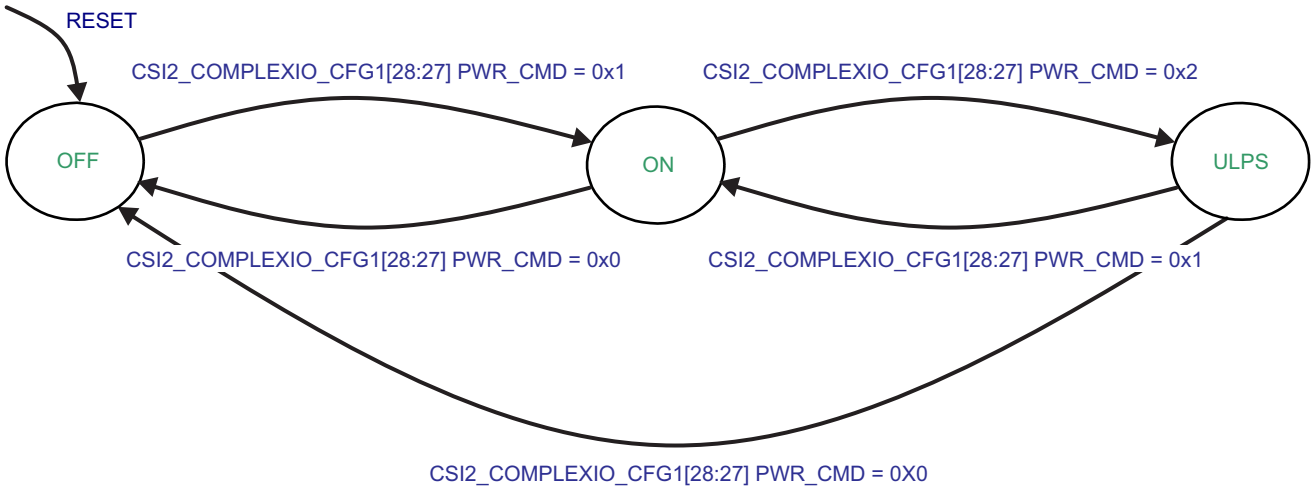
The CSI2 complex I/O can be set in three modes:

- OFF: In this power state, the complete CSI2_PHY circuit is powered down. The internal LDO is OFF.
- ON: In this power state, the complete CSI2_PHY circuit is powered on and functional.
- ULPS: Mode not supported by this module.

14.4.2.9.1.2 Complex I/O Power FSM

Figure 14-156 shows the FSM to control the power state of the complex I/O.

Figure 14-156. Complex I/O Power FSM



The PwrCmdOff, PwrCmdUlp, and PwrCmdOn commands control the state transition of the CSI2 complex I/O. The CSI2_COMPLEXIO_CFG1[28:27] PWR_CMD bit field allows the state to be modified. The allowed transitions are: OFF - ON, and ON - ULPS, and ULPS - OFF. The CSI2_COMPLEXIO_CFG1[26:25] PWR_STATUS bit field gives a status on the current state of the CSI2 complex I/O.

CAUTION

- In automatic mode, software must ensure that the CSI2 complex I/O is in ON state (that is, ON command already sent) before sending requests to the complex I/O.
- In a command request to change to a state that is the current one (acknowledge has been received), the command is ignored (nothing is sent to the CSI2 complex I/O).

14.4.2.10 CSI2 Complex I/O

14.4.2.10.1 CSI2 Complex I/O Overview

CSI2_PHY is a complex I/O with five unidirectional (HS) lane modules. This includes four data lane modules and one clock lane module. Each lane module has two data pads (DX, DY). These data pads are connected with a complementary lane module on the CSI2 receiver device using a point-to-point interconnect.

Lane modules support HS burst mode.

The maximum data rate supported in HS mode is 600 Mbps for four data lanes, and 900 Mbps for up to three data lanes. The lane module function and position are configurable; that is, any lane module can be chosen as a clock lane module, and the DX/DY data pad for each lane module can be configured as a DP or DN pin defined by the D-PHY specification.

CSI2_PHY interacts with the higher layers of the CSI2 link through the PPI. CSI2_PHY does not include a PLL; a high-frequency clock input is expected in HS mode (CLKIN4DDR). CSI2_PHY also supports the SCP interface to set various configuration and control registers.

14.4.2.10.2 Software Reset

The clock domain using the TXBYTECLKHS byte clock from the CSI2 complex I/O has dedicated reset-done information in the CSI2_COMPLEXIO_CFG1[29] RESET_DONE bit. The CSI2_SYSCONFIG[1] SOFT_RESET bit is used to reset the byte clock power domain. A dummy read using the SCP interface to any CSI2_PHY register is required after CSI2_PHY reset to complete the reset of the CSI2 complex I/O.

14.4.2.10.3 Pad Configuration

The number of lanes is configurable through the CSI2_COMPLEXIO_CFG1 register.

It is not allowed to change on the fly the position (by modifying the DATAx_POSITION, where x = 1 through 4, and CLOCK_POSITION fields), the P/N order (positive/negative order of the differential pair by modifying the DATAy_POL, where y = 1 through 4, and CLOCK_POL), or the number of active data lanes (by modifying the CSI2_COMPLEXIO_CFG1[10:8] DATA2_POSITION, CSI2_COMPLEXIO_CFG1[14:12] DATA3_POSITION, and CSI2_COMPLEXIO_CFG1[18:16] DATA4_POSITION bit fields). To add or remove lanes 2, 3, or 4, it is required to be in OFF mode for the CSI2 complex I/O.

The configuration of the CSI2 complex I/O (number of data lanes, position, differential order) must not be changed while the CSI2_CLK_CTRL[20] LP_CLK_ENABLE bit is set to 1. For hardware to recognize a new configuration of the complex I/O (done in the CSI2_COMPLEXIO_CFG1 register), the following sequence shall be followed:

1. Set the CSI2_CTRL[0] IF_EN bit to 1.
2. Reset the CSI2_CTRL[0] IF_EN bit to 0.
3. Set CSI2_CLK_CTRL[20] LP_CLK_ENABLE bit to 1.
4. Set again the CSI2_CTRL[0] IF_EN bit to 1.

If the sequence is not followed, the CSI2 complex I/O configuration is undetermined.

The minimum requirement for the number of lanes is one clock lane and one data lane. By default, data lanes 2, 3, and 4 are not connected (the reset value of the DATAx_POSITION bit fields is 0).

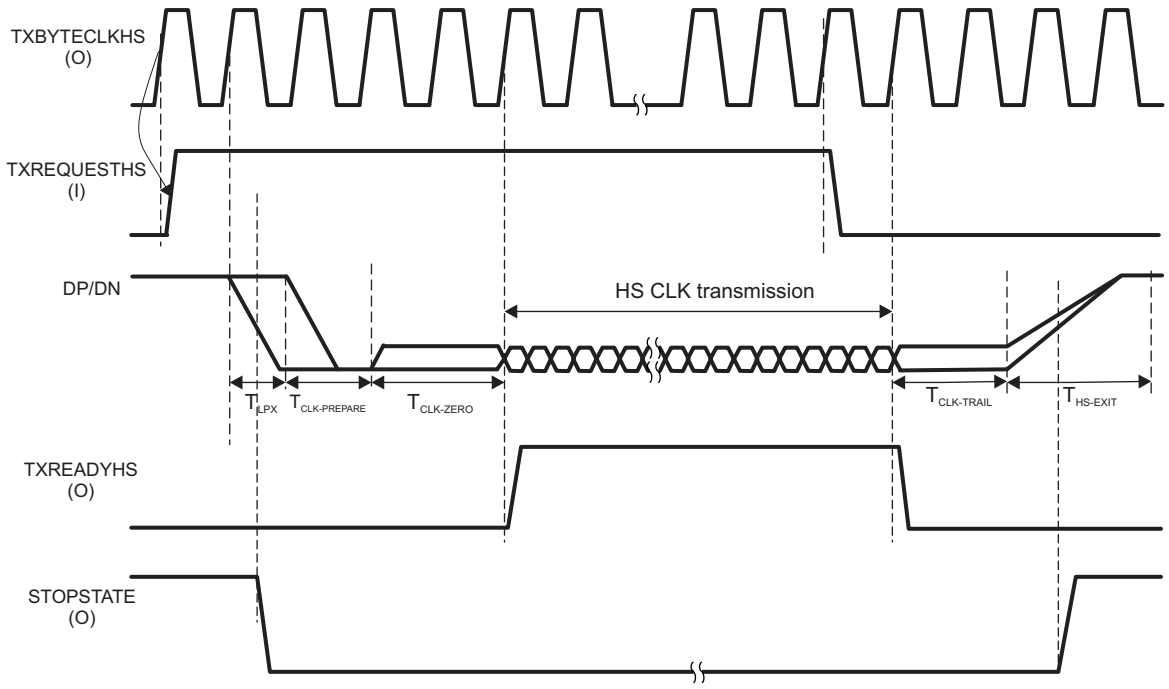
14.4.2.10.4 Display Timing Configuration

Depending on the CLKIN4DDR clock frequency settings programmed with the PRCM module, software users must program accordingly the timing parameters in the CSI2 complex I/O registers.

14.4.2.10.4.1 High-Speed Clock Transmission

Figure 14-157 shows an example of HS clock transmission.

Figure 14-157. HS Clock Transmission



TXBYTECLKHS is an output clock that is derived by dividing CLKIN4DDR by 16.
The TXBYTECLKHS clock is driven by CSI2_PHY when:

- The CLKIN4DDR is running and CSI2_PLL_CONFIGURATION2[14] PHY_CLKINEN bit is set to '1'.
- CSI2_PHY is in ON power state (CSI2_COMPLEXIO_CFG1[28:27] PWR_CMD bit = 1) and at least one lane is enabled.

To begin transmission, the protocol drives TXREQUESTHS high on a rising edge of TXBYTECLKHS. The PHY detects this signal on the next rising edge, after which it initiates the LP SoT procedure.

During an HS clock transmission, these parameters are defined in multiples of CLKIN4DDR and are programmed by the following bit fields:

- TLPX timing is programmed by CSI2_PHY_REGISTER1[20:16] TLPXBY2.
- THS-PREPARE timing is programmed by CSI2_PHY_REGISTER0[31:24] THSPREPARE.
- TCLK-ZERO timing is programmed by CSI2_PHY_REGISTER1[7:0] TCLKZERO.

TCLK-ZERO is extended, if required, so that the entire LP SoT procedure lasts an integer number of TXBYTECLKHS cycles.

At the end of the SoT procedure, HS clock transmission begins. At the same time, TXREADYHS is made high.

To stop clock transmission, the protocol drives TXREQUESTHS low on a rising edge of TXBYTECLKHS. The CSI2_PHY detects this change in TXREQUESTHS on the next edge and stops clock transmission. TXREADYHS is made low.

The CSI2_PHY then goes through the LP EoT procedure. The TCLK-TRAIL and THS-EXIT parameters are also multiples of CLKIN4DDR and are programmed by the following bit fields:

- TCLK-TRAIL timing is programmed by CSI2_PHY_REGISTER1[15:8] TCLKTRAIL.
- THS-EXIT timing is programmed by CSI2_PHY_REGISTER0[7:0] THSEXIT.

The CSI2_PHY completes the SoT and EoT procedures, once begun, regardless of any change in the PPI signals. If TXREQUESTHS goes low during the SoT procedure, the PHY starts the EoT procedure immediately after finishing the SoT procedure and no clock is transmitted.

STOPSTATE is high whenever the line is in LP-11 state, as determined by the outputs of the low-power receivers. This signal is not synchronized with TXBYTECLKHS.

The HS clock must be present for a period of time before (TCLK-PRE) and after (TCLK-POST) HS data transmission. The protocol must ensure that these timings are met by asserting and deasserting TXREQUESTHS appropriately.

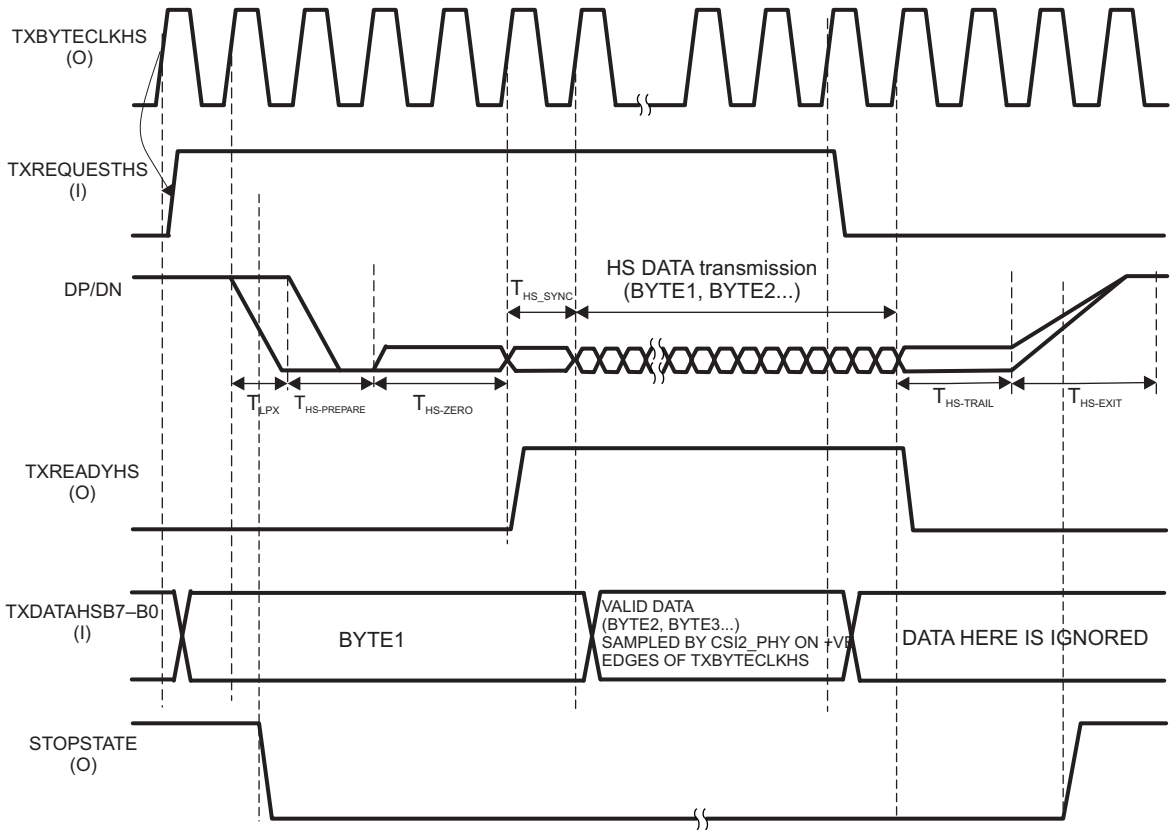
The PHY ensures that the clock signal has a quadrature phase with respect to a toggling bit sequence on any data lane, and a rising edge in the center of the first transmitted bit of each data byte. These relations are not described in the timing diagram.

CLKIN4DDR can be shut off 300 ns after the clock lane goes to STOPSTATE. Alternatively, CLKIN4DDR can be shut down after TCLK-Trail + THS-Exit + 2 Txbyteclk periods after the TxRequestHS falling edge is received by CSI2_PHY.

14.4.2.10.4.2 High-Speed Data Transmission

Figure 14-158 shows an example of HS data transmission.

Figure 14-158. HS Data Transmission



To begin transmission, the protocol drives TXDATAHS with the first byte of data on a rising edge of TXBYTECLKHS. It also makes TXREQUESTHS high on the same rising edge. The PHY detects TXREQUESTHS going high on the next rising edge of TXBYTECLKHS, following which it initiates the LP SoT procedure.

During an HS data transmission, these timings are multiples of CLKIN4DDR and are programmed by the following bit fields:

- TLPX timing is programmed by CSI2_PHY_REGISTER1[20:16] TLPXBY2.
- THS-PREPARE + THS-ZERO timing is programmed by CSI2_PHY_REGISTER0[23:16] THSPRPR_THSZERO.

THS-ZERO is extended, if required, so that the entire LP SoT procedure lasts an integer number of TXBYTECLKHS cycles. THS-SYNC corresponds to the length of the sync pattern (8 HS bits), and can be configured through the CSI2_PHY_REGISTER2[31:24] HSSYNCPATTERN bit field.

Toward the end of the SoT procedure, the PHY makes TXREADYHS high on a positive edge of TXBYTECLKHS and then starts accepting data from TXDATAHS from the next positive edge onward. The protocol is expected to provide (new) valid data on TXDATAHS on every positive edge of TXBYTECLKHS if TXREADYHS is high.

At the end of the SoT procedure, HS data transmission begins. HS data transmission happens LSB first.

To stop data transmission, the protocol drives TXREQUESTHS low on a rising edge of TXBYTECLKHS. The PHY detects this change in TXREQUESTHS on the next edge and stops data transmission. TXREADYHS is made low and data on TXDATAHS, from that point, is ignored.

The PHY then goes through the LP EoT procedure. THS-TRAIL and THS-EXIT are also multiples of CLKIN4DDR and are programmed by the following bit fields:

- THS-TRAIL timing is programmed by CSI2_PHY_REGISTER0[15:8] THSTRAIL.
- THS-EXIT timing is programmed by CSI2_PHY_REGISTER0[7:0] THSEXIT.

The PHY completes the SoT and EoT procedures, once begun, regardless of any change in PPI signals. If TXREQUESTHS goes low during the SoT procedure, the PHY starts the EoT procedure immediately after finishing the SoT procedure and no data is transmitted.

STOPSTATE is high whenever the line is in LP-11 state, as determined by the outputs of the low-power receivers. This signal is not synchronized with TXBYTECLKHS.

14.4.3 CSI2 Programming Sequence

The following sections list the programming sequence needed before the hardware triggers are generated to initiate the high-speed CSI2 data transmission.

14.4.3.1 CSI2 Global Initialization

Table 14-146. Main Sequence – PRCM and Global Configuration

Steps	Register/Bit Field/Programming	Value
Ensure the LVDS I/Os are powered off and tri-stated before CSI2 transmission	MSS_TOP_RCM. LVDSPADCTL0 MSS_TOP_RCM. LVDSPADCTL1	0x61616161 0x616161
Bring the CSI2 PE out of Idle	DSS_REG.CSICFG1.CSIMIDLEREQ	0x0
Poll to ensure the protocol engine is out of IDLE	DSS_REG.CSICFG1.CSISIDLEACK	=0x0
Enable the respective CSI2 lanes	DSS_REG.CSICFG1.CSILANEENABLE	0x-

14.4.3.2 CSI2 Protocol Engine and PHY Configuration

Table 14-147. Main Sequence – Configure CSI2 Clock Configuration

Steps	Register/Bit Field/Programming	Value
Set the ratio for the LP clock	CSI2_CLK_CTRL.LP_CLK_DIVISOR	0XA
Enable the LP clock	CSI2_CLK_CTRL.LP_CLK_ENABLE	0x1
Enable or disable clock gating to CSI2_PHY module	CSI2_CLK_CTRL.CIO_CLK_ICG	0x1
Configure if the DDR clock is also sent when no HS packets are sent to the peripheral	CSI2_CLK_CTRL.DDR_CLK_ALWAYS_ON	0x1
Enable or disable the automatic assertion and deassertion of CSI2StopClk	CSI2_CLK_CTRL.HS_AUTO_STOP_ENABLE	0x1
Configure the CSI2 functional clock synchronization and clock speed.	CSI2_CLK_CTRL.LP_RX_SYNCHRO_ENABLE	0x1
Turn ON/OFF PLL and HSDIVIDER	CSI2_CLK_CTRL.PLL_PWR_CMD	0x2

Table 14-148 summarizes the CSI2_PHY timing settings. For more information about timing calculation, see Section 14.4.2.

Table 14-148. Main Sequence – Configure CSI2_PHY

Steps	Register/Bit Field/Programming	Value
Settings of the CSI2 protocol timing. For a complete description of timing specifications see Section 14.4.2.4	CSI2_PHY_REGISTER0[31:24] REG_THSPREPARE	ceil(70ns/DDR clock period) + 2
	CSI2_PHY_REGISTER0[23:16] REG_THSPRPR_THSZERO	ceil(175ns/DDR clock period) + 2
	CSI2_PHY_REGISTER0[7:0] REG_THSEXIT	ceil(145ns/DDR clock period)
	CSI2_PHY_REGISTER0[15:8] REG_THSTRAIL	ceil(60ns/DDR clock period) + 5
	CSI2_PHY_REGISTER2[7:0] REG_TCLKPREPARE	ceil(65ns/DDR clock period)
	CSI2_PHY_REGISTER1[7:0] REG_TCLKZERO	ceil(265ns/DDR clock period)
	CSI2_PHY_REGISTER1[15:8] REG_TCLKTRAIL	ceil(60ns/DDR clock period) + 2
	CSI2_PHY_REGISTER1[20:16] REG_TLPXBY2	ceil(25ns/DDR clock period)

Table 14-149. Main Sequence – CSI2 Complex IO Configuration

Steps	Register/Bit Field/Programming	Value
Set the complex I/O power state	CSI2_COMPLEXIO_CFG1.PWR_CMD	0x1
Set the position of data lane 1	CSI2_COMPLEXIO_CFG1.DATA1_POSITION	X
Set the position of data lane 2	CSI2_COMPLEXIO_CFG1.DATA2_POSITION	X
Set the position of data lane 3	CSI2_COMPLEXIO_CFG1.DATA3_POSITION	X
Set the position of data lane 4	CSI2_COMPLEXIO_CFG1.DATA4_POSITION	X
Set the position of the clock lane	CSI2_COMPLEXIO_CFG1.CLOCK_POSITION	X
Wait until power control is ON	CSI2_COMPLEXIO_CFG1.PWR_STATUS	=0x1
Enable the CSI protocol engine	CSI2_CTRL.IF_EN	0x1
Disable the CSI protocol engine	CSI2_CTRL.IF_EN	0x0
Wait until the interface is disabled	CSI2_CTRL.IF_EN	=0x0

Table 14-150. Main Sequence – VC and OCP Configuration

Steps	Register/Bit Field/Programming	Value
Set size of the TX FIFO allocated for used VC0	CSI2_TX_FIFO_VC_SIZE	X
Set the slave interface power management mode	CSI2_SYSCONFIG.SIDLEMODE	0x1
Set the wake-up mode enable	CSI2_SYSCONFIG.ENWAKEUP	0x1
Set the OCP clock to be free-running	CSI2_SYSCONFIG.AUTO_IDLE	0x0
Set the OCP data bus width	CSI2_VC_CTRL.OCP_DATA_BUS_WIDTH	0x3
Set the trigger reset mode	CSI2_CTRL.TRIGGER_RESET_MODE	0x1
Select source of data to be OCP	CSI2_VC_CTRL.SOURCE	0x0
Configure the mode to command mode	CSI2_VC_CTRL.MODE	0x0
Enable the checksum generation for the transmit payload	CSI2_VC_CTRL.CS_TX_EN	0x1
Enable or disable the ECC generation for the transmit header	CSI2_VC_CTRL.ECC_TX_EN	0x1
Set the number of TXBYTECLKHS clock cycles	CSI2_CLK_TIMING.DDR_CLK_PRE CSI2_CLK_TIMING.DDR_CLK_POST	X
Disable the multiplication factor for the number of CSI2_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO bit-field	CSI2_TIMING1.STOP_STATE_X16_IO CSI2_TIMING1.STOP_STATE_X4_IO	0x0
Force TX stop mode	CSI2_TIMING1.FORCE_TX_STOP_MODE_IO	0x1
Enable used VC	CSI2_VC_CTRL.VC_EN	0x1
Enable the CSI protocol engine	CSI2_CTRL.IF_EN	0x1
Wait until the end of TX stop mode assertion	CSI2_TIMING1.FORCE_TX_STOP_MODE_IO	=0x1

CSI2 protocol engine VC0 FIFO size configuration requirements: because the CSI2 protocol engine has certain conditions on which it triggers the HSI transfer when data in the FIFO, the FIFO size is dependent on the CSI2 packet size.

- Case 1: If the packets to be transferred are larger than 128 bytes, configure the FIFO size to 32 entries (default configuration). By default, the FIFO size should be configured to 32 entries.
- Case 2: Single packet transferred per chirp. If the packet size is less than 128 byes, configure the FIFO size to 64 entries.
- Case 3: 2 packets sent per chirp. If the packet size is less than 128 bytes, configure the FIFO size to 64. If the second packet is less than 256 bytes, configure the FIFO size to 96.

14.4.3.3 CBUFF Configuration

Table 14-151. Main Sequence – CBUFF CSI2 Static Configuration

Steps	Register/Bit Field/Programming	Value
Set address of the CSI2 protocol engine short packet header address	CFG_SPHDR_ADDRESS	DSS_CSI2_PROTOCOL_ENGINE_U_BASE + CSI2_VC_SHORT_PACKET_HEADER
Set address of the CSI2 protocol engine long packet header address	CFG_LPHDR_ADDRESS	DSS_CSI2_PROTOCOL_ENGINE_U_BASE + CSI2_VC_LONG_PACKET_HEADER

Table 14-151. Main Sequence – CBUFF CSI2 Static Configuration (continued)

Steps	Register/Bit Field/Programming	Value
Set address of the CSI2 protocol engine long packet payload address	CFG_LPPYLD_ADDRESS	DSS_CSI2_PROTOCOL_ENGINE_U_BASE + CSI2_VC_LONG_PACKET_PAYLOAD
Configure the FIFO free threshold	CFG_FIFO_FREE_THRESHOLD	0x8
Configure the CSI2 VSYNC start code	CFG_CMD_VEVAL	0xEC000000
Configure the CSI2 VSYNC end code	CFG_CMD_VEVAL	0xEC000001
Configure the CSI2 HSYNC start code	CFG_CMD_VSVAL	0xEC000002
Configure the CSI2 HSYNC end code	CFG_CMD_VEVAL	0xEC000003
Configure the Number of chirps in a frame	CFG_CHIRPS_PER_FRAME	X
Configure the VSYNC start and end packets to be sent	CONFIG_REG_0.CVC0EN CONFIG_REG_0.CVC1EN CONFIG_REG_0.CVC2EN CONFIG_REG_0.CVC3EN	X

The configuration in Table 14-152 should be performed for each linklist entry required to transmit the CSI2 packet.

Table 14-152. Main Sequence – CBUFF Linklist

Steps	Register/Bit Field/Programming	Value
Set the valid for the linklist	CFG_DATA_LL[X].LL[X]_VALID	0x1
If linklist is the start of a new CSI2 packet, enable the long packet header	CFG_DATA_LL[X].LL[X]_LPHDR_EN	0x-
If linklist is the start of a new CSI2 packet, configure the long packet header	CFG_DATA_LL[X]_LPHDR_VAL	X
If a HSYNC start packet should be sent out before this linklist data	CFG_DATA_LL[X].LL[X]_HS	0x-
If a HSYNC end packet should be sent out at the end of this linklist data	CFG_DATA_LL[X].LL[X]_HE	0x-
Set the virtual channel number for the CSI2 packet to which the linklist belongs	CFG_DATA_LL[X].LL[X]_VCNUM	X
Configure the size in CBUFF units	CFG_DATA_LL[X].LL[X]_SIZE	X
Configure the format of the CSI2 packet to which the linklist belongs	CFG_DATA_LL[X].LL[X]_FMT	X
Set the linklist write threshold	CFG_DATA_LL[X]_THRESHOLD.LL[X]_WR_THRESHOLD	X
Set the linklist read threshold	CFG_DATA_LL[X]_THRESHOLD.LL[X]_WR_THRESHOLD	X
Set the input format	CFG_DATA_LL[X].LL[X]_FMT_IN	X

14.4.4 CSI2_PROTOCOL_ENGINE Registers

Table 14-153 lists the memory-mapped registers for the CSI2_PROTOCOL_ENGINE. All register offset addresses not listed in Table 14-153 should be considered as reserved locations and the register contents should not be modified.

Table 14-153. CSI2_PROTOCOL_ENGINE Registers

Offset	Acronym	Register Name	Section
0h	CSI2_REVISION	MODULE REVISION This register contains the IP revision code in binary coded digital. For example, we have: 0x01 = revision 0.1 and 0x21 = revision 2.1	Section 14.4.4.1
10h	CSI2_SYSCONFIG	SYSTEM CONFIGURATION REGISTER This register is the OCP-socket system configuration register.	Section 14.4.4.2
14h	CSI2_SYSSTATUS	SYSTEM STATUS REGISTER This register provides status information about the module, excluding the interrupt status register.	Section 14.4.4.3
18h	CSI2_IRQSTATUS	INTERRUPT STATUS REGISTER - All virtual channels + Complex IO + PLL This register associates one bit for each virtual channel in order to determine which virtual channel has generated the interrupt. The virtual channel shall be enabled for events to be generated on that virtual channel. If the virtual channel is disabled, the interrupt is not generated.	Section 14.4.4.4
1Ch	CSI2_IRQENABLE	INTERRUPT ENABLE REGISTER - This register associates one bit for each virtual channel in order to enable/disable each virtual channel individually.	Section 14.4.4.5
40h	CSI2_CTRL	GLOBAL CONTROL REGISTER This register controls the CSI2 Protocol Engine module. This register shall not be modified dynamically (except IF_EN bit fields).	Section 14.4.4.6
44h	CSI2_GNQ	GENERIC PARAMETER REGISTER This register provide a way to read the generic parameters used in the design.	Section 14.4.4.7
48h	CSI2_COMPLEXIO_CFG1	COMPLEXIO CONFIGURATION REGISTER for the complex IO This register contains the lane configuration for the order and position of the lanes (clock and data) and the polarity order for the control of the PHY differential signals in addition to the control bit for the power FSM.	Section 14.4.4.8
4Ch	CSI2_COMPLEXIO_IRQSTATUS	INTERRUPT STATUS REGISTER - All errors from complex IO	Section 14.4.4.9
50h	CSI2_COMPLEXIO_IRQENABLE	INTERRUPT ENABLE REGISTER - All errors from complex IO	Section 14.4.4.10
54h	CSI2_CLK_CTRL	CLOCK CONTROL This register controls the CLOCK GENERATION. The register can be modified only when IF_EN is reset.	Section 14.4.4.11
58h	CSI2_TIMING1	TIMING1 REGISTER This register controls the CSI2 Protocol Engine module timers. Any bit-field can be modified while CSI2_CTRL.IF_EN is set to '1'. It is used to indicate the number of CSI2_CLK functional clock cycles for the timers FORCE_TX_STOP_TIMER and TA_TO_TIMER	Section 14.4.4.12
5Ch	CSI2_TIMING2	TIMING2 REGISTER This register controls the CSI2 Protocol Engine module timers. Any bit-field can be modified while CSI2_CTRL.IF_EN is set to '1'. It is used to indicate the number of CSI2_CLK functional clock cycles for the timers HS_TX_TIMER and LP_RX_TIMER	Section 14.4.4.13
60h	CSI2_VM_TIMING1	VIDEO MODE TIMING REGISTER This register defines the video mode timing.	Section 14.4.4.14

Table 14-153. CSI2_PROTOCOL_ENGINE Registers (continued)

Offset	Acronym	Register Name	Section
64h	CSI2_VM_TIMING2	VIDEO MODE TIMING REGISTER This register defines the video mode timing.	Section 14.4.4.15
68h	CSI2_VM_TIMING3	VIDEO MODE TIMING REGISTER This register defines the video mode timing.	Section 14.4.4.16
6Ch	CSI2_CLK_TIMING	CLOCK TIMING REGISTER This register controls the CSI2 Protocol Engine module timers. This register shall not be modified while CSI2_CTRL.IF_EN is set to '1'.	Section 14.4.4.17
70h	CSI2_TX_FIFO_VC_SIZE	Defines the corresponding memory entries allocated for each virtual channel. The virtual channel shall be disabled in order to allocate/un-allocate some entries in the TX FIFO.	Section 14.4.4.18
74h	CSI2_RX_FIFO_VC_SIZE	Defines the corresponding memory entries allocated for each virtual channel and the addresses. The virtual channel shall be disabled in order to allocate/un-allocate some entries in the RX FIFO.	Section 14.4.4.19
78h	CSI2_COMPLEXIO_CFG2	COMPLEXIO CONFIGURATION REGISTER for the complex IO This register contains the lane configuration for the ULPS for each lane.	Section 14.4.4.20
7Ch	CSI2_RX_FIFO_VC_FULLNESS	Defines the fullness of each space allocated for each virtual channel.	Section 14.4.4.21
80h	CSI2_VM_TIMING4	VIDEO MODE TIMING REGISTER This register defines the video mode timing.	Section 14.4.4.22
84h	CSI2_TX_FIFO_VC_EMPTINESS	Defines the emptiness of each space allocated for each virtual channel.	Section 14.4.4.23
88h	CSI2_VM_TIMING5	VIDEO MODE TIMING REGISTER This register defines the video mode timing.	Section 14.4.4.24
8Ch	CSI2_VM_TIMING6	VIDEO MODE TIMING REGISTER This register defines the video mode timing.	Section 14.4.4.25
90h	CSI2_VM_TIMING7	Defines the minimum number of HS bytes clock cycles that are required to allow for the delays in entering and exiting HS mode. The supported values are from 0 to 65535	Section 14.4.4.26
94h	CSI2_STOPCLK_TIMING	Number of functional clock cycles to wait for TxByteClkHS to stop/start after change in CSI2StopClk signal	Section 14.4.4.27
98h	CSI2_CTRL2	Additional control bits for use with Video Port 2	Section 14.4.4.28
9Ch	CSI2_VM_TIMING8	VIDEO MODE TIMING REGISTER This register defines the video mode timing.	Section 14.4.4.29
A0h	CSI2_TE_HSYNC_WIDTH_0	The register configures the TE HSYNC minimum pulse width for TE0 and TE1 CMOS signals The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.	Section 14.4.4.30
A4h	CSI2_TE_VSYNC_WIDTH_0	The register configures the TE VSYNC minimum pulse width for TE0 and TE1 CMOS signals The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.	Section 14.4.4.31
A8h	CSI2_TE_HSYNC_NUMBER_0	The register configures the number of HSYNC to synchronize the beginning of the transfer on CSI2 link based on the number of HSYNC pulse received on the TE line. The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.	Section 14.4.4.32
A6h	CSI2_TE_HSYNC_WIDTH_1	The register configures the TE HSYNC minimum pulse width for TE0 and TE1 CMOS signals The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.	Section 14.4.4.30

Table 14-153. CSI2_PROTOCOL_ENGINE Registers (continued)

Offset	Acronym	Register Name	Section
AAh	CSI2_TE_VSYNC_WIDTH_1	The register configures the TE VSYNC minimum pulse width for TE0 and TE1 CMOS signals The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.	Section 14.4.4.31
AEh	CSI2_TE_HSYNC_NUMBER_1	The register configures the number of HSYNC to synchronize the beginning of the transfer on CSI2 link based on the number of HSYNC pulse received on the TE line. The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.	Section 14.4.4.32
100h	CSI2_VC_CTRL_0	CONTROL REGISTER - Virtual channel This register controls the virtual channel.	Section 14.4.4.33
104h	CSI2_VC_TE_0	CONTROL REGISTER - Virtual channel This register controls the tearing effect logic. It defines the size of the transfer when TE occurs and enables the automatic TE mode.	Section 14.4.4.34
108h	CSI2_VC_LONG_PACKET_HEADER_0	LONG PACKET HEADER INFORMATION -Virtual channel This register sets the 32-bit DATA_ID + Word count + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] WC is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.35
10Ch	CSI2_VC_LONG_PACKET_PAYLOAD_0	LONG PACKET PAYLOAD INFORMATION -Virtual channel This register sets the payload information (excluding Check-sum). The HW shall capture the word count in the packet header (in CSI2_VC_LONG_PACKET_HEADER) in order to determine the last valid data. (the virtual channel id can be different than VC). Byte1 is bit[7:0] Byte2 is bit[15:8] Byte3 is bit[23:16] Byte4 is bit[31:24] Byten is sent before Byten+1 (Least significant byte first and least significant bit first)	Section 14.4.4.36
110h	CSI2_VC_SHORT_PACKET_HEADER_0	SHORT PACKET HEADER INFORMATION -Virtual channel This register sets the 24-bit DATA_ID + Short Packet Data Field + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] Short Packet Data field is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.37
118h	CSI2_VC_IRQSTATUS_0	INTERRUPT STATUS REGISTER - Virtual channel This register regroups all the events related to the virtual channel.	Section 14.4.4.38
11Ch	CSI2_VC_IRQENABLE_0	INTERRUPT ENABLE REGISTER - Virtual channel This register regroups all the events related to virtual channel.	Section 14.4.4.39
108h	CSI2_VC_CTRL_1	CONTROL REGISTER - Virtual channel This register controls the virtual channel.	Section 14.4.4.33
10Ch	CSI2_VC_TE_1	CONTROL REGISTER - Virtual channel This register controls the tearing effect logic. It defines the size of the transfer when TE occurs and enables the automatic TE mode.	Section 14.4.4.34

Table 14-153. CSI2_PROTOCOL_ENGINE Registers (continued)

Offset	Acronym	Register Name	Section
110h	CSI2_VC_LONG_PACKET_HEADER_1	LONG PACKET HEADER INFORMATION -Virtual channel This register sets the 32-bit DATA_ID + Word count + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] WC is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.35
114h	CSI2_VC_LONG_PACKET_PAYLOAD_1	LONG PACKET PAYLOAD INFORMATION -Virtual channel This register sets the payload information (excluding Check-sum). The HW shall capture the word count in the packet header (in CSI2_VC_LONG_PACKET_HEADER) in order to determine the last valid data. (the virtual channel id can be different than VC). Byte1 is bit[7:0] Byte2 is bit[15:8] Byte3 is bit[23:16] Byte4 is bit[31:24] Byten is sent before Byten+1 (Least significant byte first and least significant bit first)	Section 14.4.4.36
118h	CSI2_VC_SHORT_PACKET_HEADER_1	SHORT PACKET HEADER INFORMATION -Virtual channel This register sets the 24-bit DATA_ID + Short Packet Data Field + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] Short Packet Data field is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.37
120h	CSI2_VC_IRQSTATUS_1	INTERRUPT STATUS REGISTER - Virtual channel This register regroups all the events related to the virtual channel.	Section 14.4.4.38
124h	CSI2_VC_IRQENABLE_1	INTERRUPT ENABLE REGISTER - Virtual channel This register regroups all the events related to virtual channel.	Section 14.4.4.39
110h	CSI2_VC_CTRL_2	CONTROL REGISTER - Virtual channel This register controls the virtual channel.	Section 14.4.4.33
114h	CSI2_VC_TE_2	CONTROL REGISTER - Virtual channel This register controls the tearing effect logic. It defines the size of the transfer when TE occurs and enables the automatic TE mode.	Section 14.4.4.34
118h	CSI2_VC_LONG_PACKET_HEADER_2	LONG PACKET HEADER INFORMATION -Virtual channel This register sets the 32-bit DATA_ID + Word count + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] WC is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.35
11Ch	CSI2_VC_LONG_PACKET_PAYLOAD_2	LONG PACKET PAYLOAD INFORMATION -Virtual channel This register sets the payload information (excluding Check-sum). The HW shall capture the word count in the packet header (in CSI2_VC_LONG_PACKET_HEADER) in order to determine the last valid data. (the virtual channel id can be different than VC). Byte1 is bit[7:0] Byte2 is bit[15:8] Byte3 is bit[23:16] Byte4 is bit[31:24] Byten is sent before Byten+1 (Least significant byte first and least significant bit first)	Section 14.4.4.36

Table 14-153. CSI2_PROTOCOL_ENGINE Registers (continued)

Offset	Acronym	Register Name	Section
120h	CSI2_VC_SHORT_PACKET_HEADER_2	SHORT PACKET HEADER INFORMATION -Virtual channel This register sets the 24-bit DATA_ID + Short Packet Data Field + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] Short Packet Data field is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.37
128h	CSI2_VC_IRQSTATUS_2	INTERRUPT STATUS REGISTER - Virtual channel This register regroups all the events related to the virtual channel.	Section 14.4.4.38
12Ch	CSI2_VC_IRQENABLE_2	INTERRUPT ENABLE REGISTER - Virtual channel This register regroups all the events related to virtual channel.	Section 14.4.4.39
118h	CSI2_VC_CTRL_3	CONTROL REGISTER - Virtual channel This register controls the virtual channel.	Section 14.4.4.33
11Ch	CSI2_VC_TE_3	CONTROL REGISTER - Virtual channel This register controls the tearing effect logic. It defines the size of the transfer when TE occurs and enables the automatic TE mode.	Section 14.4.4.34
120h	CSI2_VC_LONG_PACKET_HEADER_3	LONG PACKET HEADER INFORMATION -Virtual channel This register sets the 32-bit DATA_ID + Word count + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] WC is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.35
124h	CSI2_VC_LONG_PACKET_PAYLOAD_3	LONG PACKET PAYLOAD INFORMATION -Virtual channel This register sets the payload information (excluding Check-sum). The HW shall capture the word count in the packet header (in CSI2_VC_LONG_PACKET_HEADER) in order to determine the last valid data. (the virtual channel id can be different than VC). Byte1 is bit[7:0] Byte2 is bit[15:8] Byte3 is bit[23:16] Byte4 is bit[31:24] Byten is sent before Byten+1 (Least significant byte first and least significant bit first)	Section 14.4.4.36
128h	CSI2_VC_SHORT_PACKET_HEADER_3	SHORT PACKET HEADER INFORMATION -Virtual channel This register sets the 24-bit DATA_ID + Short Packet Data Field + ECC (the virtual channel id can be different than VC). The ECC will be computed if ECC_TX_EN is set to 1. DATA_ID is located at bit[7:0] Short Packet Data field is located at bit[23:8] ECC is located at bit[31:24] (Least significant byte first and least significant bit first)	Section 14.4.4.37
130h	CSI2_VC_IRQSTATUS_3	INTERRUPT STATUS REGISTER - Virtual channel This register regroups all the events related to the virtual channel.	Section 14.4.4.38
134h	CSI2_VC_IRQENABLE_3	INTERRUPT ENABLE REGISTER - Virtual channel This register regroups all the events related to virtual channel.	Section 14.4.4.39

Complex bit access types are encoded to fit into small table cells. Table 14-154 shows the codes that are used for access types in this section.

Table 14-154. CSI2_PROTOCOL_ENGINE Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	-0 R	Returns 0s Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

14.4.4.1 CSI2_REVISION Register (Offset = 0h) [reset = 30h]

CSI2_REVISION is shown in Figure 14-159 and described in Table 14-155.

Return to Summary Table.

MODULE REVISION This register contains the IP revision code in binary coded digital. For example, we have: 0x01 = revision 0.1 and 0x21 = revision 2.1

Figure 14-159. CSI2_REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								REV							
R-0-0h																								R-30h							

Table 14-155. CSI2_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	
7-0	REV	R	30h	IP revision [7:4] Major revision [3:0] Minor revision

14.4.4.2 CSI2_SYSCONFIG Register (Offset = 10h) [reset = 11h]

CSI2_SYSCONFIG is shown in Figure 14-160 and described in Table 14-156.

Return to Summary Table.

SYSTEM CONFIGURATION REGISTER This register is the OCP-socket system configuration register.

Figure 14-160. CSI2_SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED						CLOCKACTIVITY	
R-0-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED			SIDLEMODE		ENWAKEUP	SOFT_RESET	AUTO_IDLE
R-0-0h			R/W-2h		R/W-0h	R/W-0h	R/W-1h

Table 14-156. CSI2_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	
9-8	CLOCKACTIVITY	R/W	0h	Clocks activity during wake up mode period 0h (R/W) = OCP and Functional clocks can be switched off 1h (R/W) = Functional clocks can be switched off and OCP clocks are maintained during wake up period 2h (R/W) = OCP clocks can be switched off and Functional clocks are maintained during wake up period 3h (R/W) = OCP and Functional clocks are maintained during wake up period
7-5	RESERVED	R-0	0h	
4-3	SIDLEMODE	R/W	2h	Slave interface power management, Idle req/ack control 0h (R/W) = Force-idle. An idle request is acknowledged unconditionally 1h (R/W) = No-idle. An idle request is never acknowledged 2h (R/W) = Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module. 3h (R/W) = Reserved
2	ENWAKEUP	R/W	0h	Wake-up mode enable bit 0h (R/W) = Wakeup is disabled 1h (R/W) = Wakeup is enabled
1	SOFT_RESET	R/W	0h	Software reset. Set the bit to 1 to trigger a module reset. The bit is automatically reset by the hw. During reads return 0. 0h (R/W) = Normal mode. 1h (R/W) = The module is reset
0	AUTO_IDLE	R/W	1h	Internal OCP gating strategy 0h (R/W) = OCP clock is free-running. 1h (R/W) = Automatic OCP clock gating strategy is applied based on the OCP interface activity.

14.4.4.3 CSI2_SYSSTATUS Register (Offset = 14h) [reset = 1h]

CSI2_SYSSTATUS is shown in Figure 14-161 and described in Table 14-157.

Return to Summary Table.

SYSTEM STATUS REGISTER This register provides status information about the module, excluding the interrupt status register.

Figure 14-161. CSI2_SYSSTATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							RESET_DONE
R-0-0h							R-1h

Table 14-157. CSI2_SYSSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	
0	RESET_DONE	R	1h	Internal reset monitoring 0h (R) = Internal module reset is on going. 1h (R) = Reset completed.

14.4.4.4 CSI2_IRQSTATUS Register (Offset = 18h) [reset = 0h]

CSI2_IRQSTATUS is shown in Figure 14-162 and described in Table 14-158.

Return to Summary Table.

INTERRUPT STATUS REGISTER - All virtual channels + Complex IO + PLL This register associates one bit for each virtual channel in order to determine which virtual channel has generated the interrupt. The virtual channel shall be enabled for events to be generated on that virtual channel. If the virtual channel is disabled, the interrupt is not generated.

Figure 14-162. CSI2_IRQSTATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED	TE1_LINE_IRQ	TE0_LINE_IRQ	TA_TO_IRQ	LDO_POWER_GOOD_IRQ	SYNC_LOST_IRQ	ACK_TRIGGER_IRQ	TE_TRIGGER_IRQ
R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
LP_RX_TO_IRQ	HS_TX_TO_IRQ	RESERVED			COMPLEXIO_ERR_IRQ	PLL_RECAL_IRQ	PLL_UNLOCK_IRQ
R/W-0h	R/W-0h	R-0-0h			R-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PLL_LOCK_IRQ	RESERVED	RESYNCHRONIZATION_IRQ	WAKEUP_IRQ	VIRTUAL_CHANNEL3_IRQ	VIRTUAL_CHANNEL2_IRQ	VIRTUAL_CHANNEL1_IRQ	VIRTUAL_CHANNEL0_IRQ
R/W-0h	R-0-0h	R/W-0h	R/W-0h	R-0h	R-0h	R-0h	R-0h

Table 14-158. CSI2_IRQSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R-0	0h	
22	TE1_LINE_IRQ	R/W	0h	The VSYNC and corresponding HSYNC pulses defined in CSI2_TE_HSYNC_NUMBER for the line TE1 have been received by the CSI2 protocol engine and have trigger the start of the data transfer to the peripheral. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
21	TE0_LINE_IRQ	R/W	0h	The VSYNC and corresponding HSYNC pulses defined in CSI2_TE_HSYNC_NUMBER for the line TE0 have been received by the CSI2 protocol engine and have trigger the start of the data transfer to the peripheral. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
20	TA_TO_IRQ	R/W	0h	Turn-around Time out. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
19	LDO_POWER_GOOD_IRQ	R/W	0h	Transition of the status signal LDOPWRGOOD from the CSI2PHY indicating a state change for the supply VDDALDOCSI2PLL from up to down or down to up. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.

Table 14-158. CSI2_IRQSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	SYNC_LOST_IRQ	R/W	0h	Synchronization with Video port is lost (Video mode only) 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
17	ACK_TRIGGER_IRQ	R/W	0h	Acknowledge Trigger 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
16	TE_TRIGGER_IRQ	R/W	0h	Tearing Effect Trigger 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
15	LP_RX_TO_IRQ	R/W	0h	Interrupt for Low Power Rx Time out 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
14	HS_TX_TO_IRQ	R/W	0h	Interrupt for High Speed Tx Time out. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
13-11	RESERVED	R-0	0h	
10	COMPLEXIO_ERR_IRQ	R	0h	Error signaling from Complex IO: status of the complex IO errors received from the complex IO(events are defined in CSI2_COMPLEXIO_IRQSTATUS). 0h (R) = READS: Event is false. 1h (R) = READS: Event is true (pending).
9	PLL_RECAL_IRQ	R/W	0h	PLL recal event (assertion of CSI2Recal signal from the CSI2 PLL Control module) 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
8	PLL_UNLOCK_IRQ	R/W	0h	PLL un-clock event (de-assertion of CSI2Lock signal from the CSI2 PLL Control module) 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
7	PLL_LOCK_IRQ	R/W	0h	PLL clock event (assertion of CSI2Lock signal from the CSI2 PLL Control module) 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
6	RESERVED	R-0	0h	
5	RESYNCHRONIZATION_IRQ	R/W	0h	Video mode resynchronization indicates to the software users that the video port works but the configuration of the timings for the display controller (DISPC) and for CSI2 Protocol engine may need to be modified to avoid the resynchronization to occur. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
4	WAKEUP_IRQ	R/W	0h	Wakeup 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.

Table 14-158. CSI2_IRQSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	VIRTUAL_CHANNEL3_IRQ	R	0h	Virtual channel #3 0h (R) = READS: Event is false. 1h (R) = READS: Event is true (pending).
2	VIRTUAL_CHANNEL2_IRQ	R	0h	Virtual channel #2 0h (R) = READS: Event is false. 1h (R) = READS: Event is true (pending).
1	VIRTUAL_CHANNEL1_IRQ	R	0h	Virtual channel #1 0h (R) = READS: Event is false. 1h (R) = READS: Event is true (pending).
0	VIRTUAL_CHANNEL0_IRQ	R	0h	Virtual channel #0 0h (R) = READS: Event is false. 1h (R) = READS: Event is true (pending).

14.4.4.5 CSI2_IRQENABLE Register (Offset = 1Ch) [reset = 0h]

CSI2_IRQENABLE is shown in Figure 14-163 and described in Table 14-159.

Return to Summary Table.

INTERRUPT ENABLE REGISTER - This register associates one bit for each virtual channel in order to enable/disable each virtual channel individually.

Figure 14-163. CSI2_IRQENABLE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED	TE1_LINE_IRQ_EN	TE0_LINE_IRQ_EN	TA_TO_IRQ_EN	LDO_POWER_GOOD_IRQ_EN	SYNC_LOST_IRQ_EN	ACK_TRIGGER_IRQ_EN	TE_TRIGGER_IRQ_EN
R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
LP_RX_TO_IRQ_EN	HS_TX_TO_IRQ_EN	RESERVED				PLL_RECAL_IRQ_EN	PLL_UNLOCK_IRQ_EN
R/W-0h	R/W-0h	R-0-0h				R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PLL_LOCK_IRQ_EN	RESERVED	RESYNCHRONIZATION_IRQ_EN	WAKEUP_IRQ_EN	RESERVED			
R/W-0h	R-0-0h	R/W-0h	R/W-0h	R-0-0h			

Table 14-159. CSI2_IRQENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R-0	0h	
22	TE1_LINE_IRQ_EN	R/W	0h	The VSYNC and corresponding HSYNC pulses defined in CSI2_TE_HSYNC_NUMBER for the line TE1 have been received by the CSI2 protocol engine and have trigger the start of the data transfer to the peripheral. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
21	TE0_LINE_IRQ_EN	R/W	0h	The VSYNC and corresponding HSYNC pulses defined in CSI2_TE_HSYNC_NUMBER for the line TE0 have been received by the CSI2 protocol engine and have trigger the start of the data transfer to the peripheral. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
20	TA_TO_IRQ_EN	R/W	0h	Turn-around Time out. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
19	LDO_POWER_GOOD_IRQ_EN	R/W	0h	Transition of the status signal LDOPWRGOOD from the CSI2PHY indicating a state change for the supply VDDALDOCSI2PLL from up to down or down to up. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
18	SYNC_LOST_IRQ_EN	R/W	0h	Synchronization with Video port is lost (Video mode only) 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
17	ACK_TRIGGER_IRQ_EN	R/W	0h	Acknowledge trigger 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs

Table 14-159. CSI2_IRQENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	TE_TRIGGER_IRQ_EN	R/W	0h	Tearing Effect trigger 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
15	LP_RX_TO_IRQ_EN	R/W	0h	Interrupt for Low Power Rx Time out. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
14	HS_TX_TO_IRQ_EN	R/W	0h	Interrupt for High Speed Tx Time out. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
13-10	RESERVED	R-0	0h	
9	PLL_RECAL_IRQ_EN	R/W	0h	PLL recal event (assertion of CSI2Recal signal from the CSI2 PLL Control module) 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
8	PLL_UNLOCK_IRQ_EN	R/W	0h	PLL un-clock event (de-assertion of CSI2Lock signal from the CSI2 PLL Control module) 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
7	PLL_LOCK_IRQ_EN	R/W	0h	PLL clock event (assertion of CSI2Lock signal from the CSI2 PLL Control module) 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
6	RESERVED	R-0	0h	
5	RESYNCHRONIZATION_IRQ_EN	R/W	0h	Video mode resynchronization 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
4	WAKEUP_IRQ_EN	R/W	0h	Wakeup 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
3-0	RESERVED	R-0	0h	

14.4.4.6 CSI2_CTRL Register (Offset = 40h) [reset = 100h]

CSI2_CTRL is shown in Figure 14-164 and described in Table 14-160.

Return to Summary Table.

GLOBAL CONTROL REGISTER
This register controls the CSI2 Protocol Engine module. This register shall not be modified dynamically (except IF_EN bit fields).

Figure 14-164. CSI2_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							DISPC_UPDATE_SYNC
R-0-0h							R/W-0h
23	22	21	20	19	18	17	16
HSA_BLANKING_MODE	HBP_BLANKING_MODE	HFP_BLANKING_MODE	BLANKING_MODE	EOT_ENABLE	VP_HSYNC_END	VP_HSYNC_START	VP_VSYNC_END
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
VP_VSYNC_START	TRIGGER_RESET_MODE	LINE_BUFFER		VP_VSYNC_POL	VP_HSYNC_POL	VP_DE_POL	VP_CLK_POL
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-1h
7	6	5	4	3	2	1	0
VP_DATA_BUS_WIDTH		TRIGGER_RESET	VP_CLK_RATIO	TX_FIFO_ARBITRATION	ECC_RX_EN	CS_RX_EN	IF_EN
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-160. CSI2_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R-0	0h	
24	DISPC_UPDATE_SYNC	R/W	0h	Determines if the Dispc_Update_Sync signal from the display controller is used. 0h (R/W) = Dispc_Update_Sync signal is not used. 1h (R/W) = Dispc_Update_Sync signal is used.
23	HSA_BLANKING_MODE	R/W	0h	Blanking mode 0h (R/W) = Packets in TX FIFO are sent during HSA blanking period of video mode or LPS is used. 1h (R/W) = LONG BLANKING PACKETS only are used during HSA blanking period of video mode.
22	HBP_BLANKING_MODE	R/W	0h	Blanking mode 0h (R/W) = Packets in TX FIFO are sent during HBP blanking period of video mode or LPS is used. 1h (R/W) = LONG BLANKING PACKETS only are used during HBP blanking period of video mode.
21	HFP_BLANKING_MODE	R/W	0h	Blanking mode 0h (R/W) = Packets in TX FIFO are sent during HFP blanking period of video mode or LPS is used. 1h (R/W) = LONG BLANKING PACKETS only are used during HFP blanking period of video mode.

Table 14-160. CSI2_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	BLANKING_MODE	R/W	0h	Blanking mode 0h (R/W) = LPS is used during blanking periods of video mode (except HSA, HBP, HFP defined in HSA_BLANKING_MODE, HBP_BLANKING_MODE AND HFP_BLANKING_MODE respectively) when there is no command mode data in TX FIFO ready to be sent. So blanking periods can be different during the frame depending on the TX FIFO. 1h (R/W) = LONG BLANKING PACKETS are used during blanking periods of video mode (except HSA, HBP, HFP defined in HSA_BLANKING_MODE, HBP_BLANKING_MODE AND HFP_BLANKING_MODE respectively) regardless of the packets present in the TX FIFO ready to be sent
19	EOT_ENABLE	R/W	0h	Enable EOT packets at the end of HS transmission. 0h (R/W) = No EOT packets 1h (R/W) = EOT packet is sent at all HS to LP transitions.
18	VP_HSYNC_END	R/W	0h	HSYNC end pulse. 0h (R/W) = Disabled. No HSYNC END short packet is generated. 1h (R/W) = Enabled. While the HSYNC END pulse is detected, the associated short packet HSYNC END is generated.
17	VP_HSYNC_START	R/W	0h	HSYNC start pulse. 0h (R/W) = Disabled. No HSYNC START short packet is generated. 1h (R/W) = Enabled. While the HSYNC start pulse is detected, the associated short packet HSYNC START is generated.
16	VP_VSYNC_END	R/W	0h	VSYNC end pulse. 0h (R/W) = Disabled. No VSYNC END short packet is generated. 1h (R/W) = Enabled. While the VSYNC END pulse is detected, the associated short packet VSYNC END is generated.
15	VP_VSYNC_START	R/W	0h	VSYNC start pulse. 0h (R/W) = Disabled. No VSYNC START short packet is generated. 1h (R/W) = Enabled. While the VSYNC START pulse is detected, the associated short packet VSYNC START is generated.
14	TRIGGER_RESET_MODE	R/W	0h	Selection of the trigger reset mode 0h (R/W) = Synchronized: the mode is only valid if there is virtual channel using the video mode and it is active. The principal is to wait for the current video frame to be transferred on the link. Any data received after the VSYNC are ignored. 1h (R/W) = Immediate: all pending requests in TX FIFO are taken into account for transfer scheduling, the RX FIFO is ignored, and the data from video port are ignored as soon as possible. Only the current transfer on CSI2 link and already scheduled ones are transmitted. All the other transfers are discarded.
13-12	LINE_BUFFER	R/W	0h	Number of line buffers to be used while receiving data on the video port. The valid values are from 0 to CSI2_GNQ.VP1_NB_LINE_BUFFER. 0h (R/W) = No line buffer 1h (R/W) = 1 line buffer 2h (R/W) = 2 line buffers
11	VP_VSYNC_POL	R/W	0h	VP vertical synchronization signal polarity 0h (R/W) = VSYNC signal on the video port is active low. 1h (R/W) = VSYNC signal on the video port is active high.
10	VP_HSYNC_POL	R/W	0h	VP horizontal synchronization signal polarity 0h (R/W) = HSYNC signal on the video port is active low. 1h (R/W) = HSYNC signal on the video port is active high.
9	VP_DE_POL	R/W	0h	VP data enable signal polarity 0h (R/W) = DE signal on the video port is active low. 1h (R/W) = DE signal on the video port is active high.

Table 14-160. CSI2_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	VP_CLK_POL	R/W	1h	VP clock polarity 0h (R/W) = The CSI2 Protocol Engine module captures the data on the VP on the pixel clock falling edge. The module connected to the VP shall drive the data on the pixel clock rising edge. 1h (R/W) = The CSI2 Protocol Engine module captures the data on the VP on the pixel clock raising edge. The module connected to the VP shall drive the data on the pixel clock falling edge.
7-6	VP_DATA_BUS_WIDTH	R/W	0h	Defines the size of the video port data bus 0h (R/W) = 16-bits data width (LSB of the 24-bit video port data bus) 1h (R/W) = 18-bits data width (LSB of the 24-bit video port data bus) 2h (R/W) = 24-bits data width (LSB of the 24-bit video port data bus)
5	TRIGGER_RESET	R/W	0h	Send the reset trigger to the peripheral. 0h (R/W) = READS: Reset trigger generation is completed. It is reset by HW when it is completed. WRITES: Cancellation of the request for Reset trigger generation (maybe too late since it is already on going) 1h (R/W) = READS: Generation of the reset trigger has been requested by user (could be on going but not completed yet). WRITES: Request for Reset trigger to be sent to the peripheral.
4	VP_CLK_RATIO	R/W	0h	The field indicates the clock ratio between VP.CLK and VP.PCLK. The clock VP.PCLK is generated from VP.CLK. It is divided down. The information is only used when the video port is used to provide data in command mode. In the case of video mode, it is not used. 0h (R/W) = The clock VP.PCLK is the clock VP.CLK divided by 2. The duty cycle of VP.PCLK is 50/50. 1h (R/W) = The clock VP.PCLK is the clock VP.CLK divided by 3 or more. The duty cycle of VP.PCLK is not 50/50 for odd ratio numbers (3,5,7,...).
3	TX_FIFO_ARBITRATION	R/W	0h	Defines the arbitration scheme for granting the virtual channel pending ready requests in the TX FIFO 0h (R/W) = Round-Robin Scheme is used 1h (R/W) = Sequential Scheme is used
2	ECC_RX_EN	R/W	0h	Enables the Error Correction Code check for the received header (short and long packets for all virtual channel ids). 0h (R/W) = Disabled 1h (R/W) = Enabled
1	CS_RX_EN	R/W	0h	Enables the checksum check for the received payload (long packet only for all virtual channel ids). 0h (R/W) = Disabled 1h (R/W) = Enabled
0	IF_EN	R/W	0h	Enables the module. When the module is disabled the signals from the complex IO are gated (no updates of the interrupt status register). It is not possible to change the bit-fields in the register CSI2_CTRL except IF_EN when it is enabled. All the other registers can be changed except the ones that require CSI2_VC_CTRL.VC_EN to be equal to 0 to be modified. 0h (R/W) = The interface is disabled. If one of the virtual channel uses the video mode with the video port to receive the data, the CSI2 protocol engines is disabled when the next VSYNC is received and all the data in the FIFO for the other virtual channels in command mode are sent to the peripherals (if BTA_EN is enabled, the CSI2 protocol needs to wait for the response and BTA from the peripheral before disabling all the internal logic since an acknowledge is requested). 1h (R/W) = The interface is enabled immediately, the data acquisition on the video port starts on the next VSYNC (video mode) or first data received in the Slave port FIFO (command mode).

14.4.4.7 CSI2_GNQ Register (Offset = 44h) [reset = 00926936h]

CSI2_GNQ is shown in Figure 14-165 and described in Table 14-161.

Return to Summary Table.

GENERIC PARAMETER REGISTER This register provide a way to read the generic parameters used in the design.

Figure 14-165. CSI2_GNQ Register

31	30	29	28	27	26	25	24
RESERVED							NB_VIDEO_PORTS
R-0-0h							R-0h
23	22	21	20	19	18	17	16
VP2_NB_LINE_BUFFER		RESERVED	VP2_LINE_BUFFER_SIZE			VP1_NB_LINE_BUFFER	
R-2h		R-0-0h	R-4h			R-2h	
15	14	13	12	11	10	9	8
RESERVED	VP1_LINE_BUFFER_SIZE			NB_DATA_LANES			NB_DMA_REQUEST
R-0-0h	R-6h			R-4h			R-4h
7	6	5	4	3	2	1	0
NB_DMA_REQUEST		RX_FIFODEPTH			TX_FIFODEPTH		
R-4h		R-6h			R-6h		

Table 14-161. CSI2_GNQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R-0	0h	
24	NB_VIDEO_PORTS	R	0h	Number of video ports 0h (R) = Video port 1 only is present 1h (R) = Video port 1 and video port 2 are present
23-22	VP2_NB_LINE_BUFFER	R	2h	Determines the number of video buffer lines associated to video port #2. 0h (R) = No line buffer 1h (R) = 1 line buffer of the size defined in LINE_BUFFER_SIZE 2h (R) = 2 line buffers of the size defined in LINE_BUFFER_SIZE
21	RESERVED	R-0	0h	
20-18	VP2_LINE_BUFFER_SIZE	R	4h	Determines the video line buffer size associated to video port #2 . 1h (R) = 512x24-bits, 682x18-bits, 768x16bits (memory of 384x32-bits) 2h (R) = 682x24-bits, 910x18-bits, 1024x16bits (memory of 512x32-bits) 3h (R) = 853x24-bits, 1137x18-bits, 1280x16bits (memory of 640x32-bits) 4h (R) = 1024x24-bits, 1365x18-bits, 1536x16bits (memory of 768x32-bits) 5h (R) = 1194x24-bits, 1592x18-bits, 1792x16bits (memory of 896x32-bits) 6h (R) = 1365x24-bits, 1820x18-bits, 2048x16bits (memory of 1024x32-bits)
17-16	VP1_NB_LINE_BUFFER	R	2h	Determines the number of video buffer lines associated to video port #1. 0h (R) = No line buffer 1h (R) = 1 line buffer of the size defined in LINE_BUFFER_SIZE 2h (R) = 2 line buffers of the size defined in LINE_BUFFER_SIZE
15	RESERVED	R-0	0h	

Table 14-161. CSI2_GNQ Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-12	VP1_LINE_BUFFER_SIZE	R	6h	Determines the video line buffer size associated to video port #1 . 1h (R) = 512x24-bits, 682x18-bits, 768x16bits (memory of 384x32-bits) 2h (R) = 682x24-bits, 910x18-bits, 1024x16bits (memory of 512x32-bits) 3h (R) = 853x24-bits, 1137x18-bits, 1280x16bits (memory of 640x32-bits) 4h (R) = 1024x24-bits, 1365x18-bits, 1536x16bits (memory of 768x32-bits) 5h (R) = 1194x24-bits, 1592x18-bits, 1792x16bits (memory of 896x32-bits) 6h (R) = 1365x24-bits, 1820x18-bits, 2048x16bits (memory of 1024x32-bits)
11-9	NB_DATA_LANES	R	4h	Determines the number of data lanes supported by the CSI2 protocol engine . 1h (R) = 1 Data lane 2h (R) = 2 Data lanes 3h (R) = 3 Data lanes 4h (R) = 4 Data lanes
8-6	NB_DMA_REQUEST	R	4h	Determines the number of DMA_REQ signals. 0h (R) = No DMA request 1h (R) = 1 DMA request 2h (R) = 2 DMA requests 3h (R) = 3 DMA requests 4h (R) = 4 DMA requests
5-3	RX_FIFODEPTH	R	6h	Determines the data RX FIFO depth (32-bit words) on the slave port. 4h (R) = 32x 33 bits 5h (R) = 64x 33 bits 6h (R) = 128 x 33 bits 7h (R) = 256 x 33 bits
2-0	TX_FIFODEPTH	R	6h	Determines the data TX FIFO depth (33-bit words) on the slave port. 4h (R) = 32x 33 bits 5h (R) = 64x 33 bits 6h (R) = 128 x 33 bits 7h (R) = 256 x 33 bits

14.4.4.8 CSI2_COMPLEXIO_CFG1 Register (Offset = 48h) [reset = 20000000h]

CSI2_COMPLEXIO_CFG1 is shown in Figure 14-166 and described in Table 14-162.

Return to Summary Table.

COMPLEXIO CONFIGURATION REGISTER for the complex IO This register contains the lane configuration for the order and position of the lanes (clock and data) and the polarity order for the control of the PHY differential signals in addition to the control bit for the power FSM.

Figure 14-166. CSI2_COMPLEXIO_CFG1 Register

31	30	29	28	27	26	25	24
SHADOWING	GOBIT	RESET_DONE	PWR_CMD		PWR_STATUS		RESERVED
R/W-0h	R/W-0h	R-1h	R/W-0h		R-0h		R-0-0h
23	22	21	20	19	18	17	16
RESERVED		LDO_POWER_GOOD_STATE	USE_LDO_EXTERNAL	DATA4_POL	XXXX		DATA4_POSITION
R-0-0h		R-0h	R/W-0h	R/W-0h			R/W-0h
15	14	13	12	11	10	9	8
DATA3_POL	XXXX		DATA3_POSITION	DATA2_POL	RESERVED	XXXX	DATA2_POSITION
R/W-0h			R/W-0h	R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
DATA1_POL	DATA1_POSITION			CLOCK_POL	CLOCK_POSITION		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		

Table 14-162. CSI2_COMPLEXIO_CFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SHADOWING	R/W	0h	Shadowing configuration. 0h (R/W) = Disabled. The writes to the first 3 registers of the complex IO address map is done like the other SCP registers. 1h (R/W) = Enabled. The writes to the first 3 registers of the complex IO address map is done only when the GObit is set and when the signal DISPCUpdataSync from the display controller module is active.
30	GOBIT	R/W	0h	Allows the synchronized update of the shadow registers when the signal DISPCUpdateSync is active. 0h (R/W) = Resets the GOBIT. The hardware has finished the update of the shadow SCP registers. The bit is reset by Hardware. The SW can reset the bit in case the user decides to abort it. There is no guarantee that the SW reset is done before the transfer of the values to the complex IO. 1h (R/W) = Set the GOBIT. Only when the transfer of the new values for the three first registers is completed (3, 2, 1, or 0 transfers are performed based on the number of registers to update), the GOBIT is reset. The DISPCUpdateSync signal is used to synchronize the update. The bit shall be set only when it is in reset state.
29	RESET_DONE	R	1h	Internal reset monitoring of the power domain using the PPI byte clock from the complex IO 0h (R) = Internal module reset is on going. 1h (R) = Reset completed.
28-27	PWR_CMD	R/W	0h	Command for power control of the complex IO 0h (R/W) = Command to change to OFF state 1h (R/W) = Command to change to ON state 2h (R/W) = Command to change to Ultra Low Power state

Table 14-162. CSI2_COMPLEXIO_CFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26-25	PWR_STATUS	R	0h	Status of the power control of the complex IO 0h (R) = Complex IO in OFF state 1h (R) = Complex IO in ON state 2h (R) = Complex IO in Ultra Low Power state
24-22	RESERVED	R-0	0h	
21	LDO_POWER_GOOD_STATE	R	0h	Indicates the state of the signal LDOPWRGOOD. VDDALDOCSI2PLL: 1.2V power supply for the PLL. The voltage is supplied by the internal or external LDO. The interrupt LDO_POWER_GOOD_IRQ is generated when a transition is detected on the signal LDOPWRGOOD from the CSI2PHY. 0h (R) = VDDALDOCSI2PLL power supply is down 1h (R) = VDDALDOCSI2PLL power supply is up
20	USE_LDO_EXTERNAL	R/W	0h	Select the external LDO for the CSI2PHY. 0h (R/W) = CSI2PHY internal LDO is used. 1h (R/W) = External LDO is used. CSI2PHY LDO is tri-stated.
19	RESERVED	R-0	0h	
19	DATA4_POL	R/W	0h	+/- differential pin order of DATA lane 4. 0h (R/W) = +/- pin order (CSI2.DX=+ and CSI2.DY=-) 1h (R/W) = -/+ pin order (CSI2.DX=- and CSI2.DY=+)
16	RESERVED	R-0	0h	
16	DATA4_POSITION	R/W	0h	Position and order of the DATA lane 4. 0h (R/W) = Not used/connected 1h (R/W) = Data lane 4 is at the position 1. 2h (R/W) = Data lane 4 is at the position 2. 3h (R/W) = Data lane 4 is at the position 3. 4h (R/W) = Data lane 4 is at the position 4. 5h (R/W) = Data lane 4 is at the position 5.
15	RESERVED	R-0	0h	
15	DATA3_POL	R/W	0h	+/- differential pin order of DATA lane 3. 0h (R/W) = +/- pin order (CSI2.DX=+ and CSI2.DY=-) 1h (R/W) = -/+ pin order (CSI2.DX=- and CSI2.DY=+)
12	RESERVED	R-0	0h	
12	DATA3_POSITION	R/W	0h	Position and order of the DATA lane 3. 0h (R/W) = Not used/connected 1h (R/W) = Data lane 3 is at the position 1. 2h (R/W) = Data lane 3 is at the position 2. 3h (R/W) = Data lane 3 is at the position 3. 4h (R/W) = Data lane 3 is at the position 4. 5h (R/W) = Data lane 3 is at the position 5.
11	RESERVED	R-0	0h	
11	DATA2_POL	R/W	0h	+/- differential pin order of DATA lane 2. 0h (R/W) = +/- pin order (CSI2.DX=+ and CSI2.DY=-) 1h (R/W) = -/+ pin order (CSI2.DX=- and CSI2.DY=+)
10	RESERVED	R/W	0h	
8	RESERVED	R-0	0h	
8	DATA2_POSITION	R/W	0h	Position and order of the DATA lane 2. 0h (R/W) = Not used/connected 1h (R/W) = Data lane 2 is at the position 1. 2h (R/W) = Data lane 2 is at the position 2. 3h (R/W) = Data lane 2 is at the position 3. 4h (R/W) = Data lane 2 is at the position 4. 5h (R/W) = Data lane 2 is at the position 5.

Table 14-162. CSI2_COMPLEXIO_CFG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	DATA1_POL	R/W	0h	+/- pin differential pin order of DATA lane 1 0h (R/W) = +/- pin order (CSI2.DX=+ and CSI2.DY=-) 1h (R/W) = -/+ pin order (CSI2.DX=- and CSI2.DY=+)
6-4	DATA1_POSITION	R/W	0h	Position and order of the DATA lane 1. The data lane 1 is always present. 1h (R/W) = Data lane 1 is at the position 1. 2h (R/W) = Data lane 1 is at the position 2. 3h (R/W) = Data lane 1 is at the position 3. 4h (R/W) = Data lane 1 is at the position 4. 5h (R/W) = Data lane 1 is at the position 5.
3	CLOCK_POL	R/W	0h	+/- differential pin order of CLOCK lane. 0h (R/W) = +/- pin order (CSI2.DX=+ and CSI2.DY=-) 1h (R/W) = -/+ pin order (CSI2.DX=- and CSI2.DY=+)
2-0	CLOCK_POSITION	R/W	0h	Position and order of the CLOCK lane. 0, 5, 6 and 7 are reserved. The clock lane is always present but can not be at the position 5 even if the COMPLEX IO consists of 5 lanes. 1h (R/W) = Clock lane is at the position 1. 2h (R/W) = Clock lane is at the position 2. 3h (R/W) = Clock lane is at the position 3. 4h (R/W) = Clock lane is at the position 4.

14.4.4.9 CSI2_COMPLEXIO_IRQSTATUS Register (Offset = 4Ch) [reset = 0h]

CSI2_COMPLEXIO_IRQSTATUS is shown in Figure 14-167 and described in Table 14-163.

Return to Summary Table.

INTERRUPT STATUS REGISTER - All errors from complex IO

Figure 14-167. CSI2_COMPLEXIO_IRQSTATUS Register

31	30	29	28	27	26	25	24
ULPSACTIVENOT_ALL1_IRQ	ULPSACTIVENOT_ALL0_IRQ	ERRCONTENTIONLP1_5_IRQ	ERRCONTENTIONLP0_5_IRQ	ERRCONTENTIONLP1_4_IRQ	ERRCONTENTIONLP0_4_IRQ	ERRCONTENTIONLP1_3_IRQ	ERRCONTENTIONLP0_3_IRQ
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
ERRCONTENTIONLP1_2_IRQ	ERRCONTENTIONLP0_2_IRQ	ERRCONTENTIONLP1_1_IRQ	ERRCONTENTIONLP0_1_IRQ	STATEULPS5_IRQ	STATEULPS4_IRQ	STATEULPS3_IRQ	STATEULPS2_IRQ
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
STATEULPS1_IRQ	ERRCONTROL5_IRQ	ERRCONTROL4_IRQ	ERRCONTROL3_IRQ	ERRCONTROL2_IRQ	ERRCONTROL1_IRQ	ERRESC5_IRQ	ERRESC4_IRQ
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
ERRESC3_IRQ	ERRESC2_IRQ	ERRESC1_IRQ	ERRSYNCESC5_IRQ	ERRSYNCESC4_IRQ	ERRSYNCESC3_IRQ	ERRSYNCESC2_IRQ	ERRSYNCESC1_IRQ
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-163. CSI2_COMPLEXIO_IRQSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ULPSACTIVENOT_ALL1_IRQ	R/W	0h	All the ULPSActiveNOT signals corresponding to the lanes with TXULPSExit being high are high. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
30	ULPSACTIVENOT_ALL0_IRQ	R/W	0h	All signals ULPSActiveNOT are 0 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
29	RESERVED	R-0	0h	
29	ERRCONTENTIONLP1_5_IRQ	R/W	0h	Contention LP1 error for lane #5 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
28	RESERVED	R-0	0h	
28	ERRCONTENTIONLP0_5_IRQ	R/W	0h	Contention LP0 error for lane #5 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
27	RESERVED	R-0	0h	
27	ERRCONTENTIONLP1_4_IRQ	R/W	0h	Contention LP1 error for lane #4 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
26	RESERVED	R-0	0h	

Table 14-163. CSI2_COMPLEXIO_IRQSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	ERRCONTENTIONLP0_4_IRQ	R/W	0h	Contention LP0 error for lane #4 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
25	RESERVED	R-0	0h	
25	ERRCONTENTIONLP1_3_IRQ	R/W	0h	Contention LP1 error for lane #3 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
24	RESERVED	R-0	0h	
24	ERRCONTENTIONLP0_3_IRQ	R/W	0h	Contention LP0 error for lane #3 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
23	ERRCONTENTIONLP1_2_IRQ	R/W	0h	Contention LP1 error for lane #2 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
22	ERRCONTENTIONLP0_2_IRQ	R/W	0h	Contention LP0 error for lane #2 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
21	ERRCONTENTIONLP1_1_IRQ	R/W	0h	Contention LP1 error for lane #1 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
20	ERRCONTENTIONLP0_1_IRQ	R/W	0h	Contention LP0 error for lane #1 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
19	RESERVED	R-0	0h	
19	STATEULPS5_IRQ	R/W	0h	Lane #5 in Ultra Low Power State 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
18	RESERVED	R-0	0h	
18	STATEULPS4_IRQ	R/W	0h	Lane #4 in Ultra Low Power Mode 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
17	RESERVED	R-0	0h	
17	STATEULPS3_IRQ	R/W	0h	Lane #3 in Ultra Low Power State 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
16	STATEULPS2_IRQ	R/W	0h	Lane #2 in Ultra Low Power State 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
15	STATEULPS1_IRQ	R/W	0h	Lane #1 in Ultra Low Power State 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.

Table 14-163. CSI2_COMPLEXIO_IRQSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	RESERVED	R-0	0h	
14	ERRCONTROL5_IRQ	R/W	0h	Control error for lane #5 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
13	RESERVED	R-0	0h	
13	ERRCONTROL4_IRQ	R/W	0h	Control error for lane #4 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
12	RESERVED	R-0	0h	
12	ERRCONTROL3_IRQ	R/W	0h	Control error for lane #3 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
11	ERRCONTROL2_IRQ	R/W	0h	Control error for lane #2 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
10	ERRCONTROL1_IRQ	R/W	0h	Control error for lane #1 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
9	RESERVED	R-0	0h	
9	ERRESC5_IRQ	R/W	0h	Escape entry error for lane #5 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
8	RESERVED	R-0	0h	
8	ERRESC4_IRQ	R/W	0h	Escape entry error for lane #4 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
7	RESERVED	R-0	0h	
7	ERRESC3_IRQ	R/W	0h	Escape entry error for lane #3 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
6	ERRESC2_IRQ	R/W	0h	Escape entry error for lane #2 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
5	ERRESC1_IRQ	R/W	0h	Escape entry error for lane #1 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
4	RESERVED	R-0	0h	
4	ERRSYNCESC5_IRQ	R/W	0h	Low power Data transmission synchronization error for lane #5 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
3	RESERVED	R-0	0h	

Table 14-163. CSI2_COMPLEXIO_IRQSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	ERRSYNCESC4_IRQ	R/W	0h	Low power Data transmission synchronization error for lane #4 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
2	RESERVED	R-0	0h	
2	ERRSYNCESC3_IRQ	R/W	0h	Low power Data transmission synchronization error for lane #3 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
1	ERRSYNCESC2_IRQ	R/W	0h	Low power Data transmission synchronization error for lane #2 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
0	ERRSYNCESC1_IRQ	R/W	0h	Low power Data transmission synchronization error for lane #1 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.

14.4.4.10 CSI2_COMPLEXIO_IRQENABLE Register (Offset = 50h) [reset = 0h]

CSI2_COMPLEXIO_IRQENABLE is shown in Figure 14-168 and described in Table 14-164.

Return to Summary Table.

INTERRUPT ENABLE REGISTER - All errors from complex IO

Figure 14-168. CSI2_COMPLEXIO_IRQENABLE Register

31	30	29	28	27	26	25	24
ULPSACTIVENOT_ALL1_IRQ_EN	ULPSACTIVENOT_ALL0_IRQ_EN	RESERVED	ERRCONTENTIONLP0_5_IRQ_EN	ERRCONTENTIONLP1_4_IRQ_EN	ERRCONTENTIONLP0_4_IRQ_EN	ERRCONTENTIONLP1_3_IRQ_EN	ERRCONTENTIONLP0_3_IRQ_EN
R/W-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
ERRCONTENTIONLP1_2_IRQ_EN	ERRCONTENTIONLP0_2_IRQ_EN	ERRCONTENTIONLP1_1_IRQ_EN	ERRCONTENTIONLP0_1_IRQ_EN	STATEULPS5_IRQ_EN	STATEULPS4_IRQ_EN	STATEULPS3_IRQ_EN	STATEULPS2_IRQ_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
STATEULPS1_IRQ_EN	ERRCONTROL5_IRQ_EN	ERRCONTROL4_IRQ_EN	ERRCONTROL3_IRQ_EN	ERRCONTROL2_IRQ_EN	ERRCONTROL1_IRQ_EN	ERRESC5_IRQ_EN	ERRESC4_IRQ_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
ERRESC3_IRQ_EN	ERRESC2_IRQ_EN	ERRESC1_IRQ_EN	ERRSYNCSESC5_IRQ_EN	ERRSYNCSESC4_IRQ_EN	ERRSYNCSESC3_IRQ_EN	ERRSYNCSESC2_IRQ_EN	ERRSYNCSESC1_IRQ_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-164. CSI2_COMPLEXIO_IRQENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ULPSACTIVENOT_ALL1_IRQ_EN	R/W	0h	All the ULPSActiveNOT signals corresponding to the lanes with TXULPSExit being high are high. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
30	ULPSACTIVENOT_ALL0_IRQ_EN	R/W	0h	All signals ULPSActiveNOT are 0 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
29	ERRCONTENTIONLP1_5_IRQ_EN	R/W	0h	Contention LP1 error for lane #5 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
29	RESERVED	R-0	0h	Write 0's for future compatibility. Reads returns 0.
28	RESERVED	R-0	0h	Write 0's for future compatibility. Reads returns 0.
28	ERRCONTENTIONLP0_5_IRQ_EN	R/W	0h	Contention LP0 error for lane #5 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
27	RESERVED	R-0	0h	Write 0's for future compatibility. Reads returns 0.
27	ERRCONTENTIONLP1_4_IRQ_EN	R/W	0h	Contention LP1 error for lane #4 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
26	RESERVED	R-0	0h	Write 0's for future compatibility. Reads returns 0.

Table 14-164. CSI2_COMPLEXIO_IRQENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	ERRCONTENTIONLP0_4_IRQ_EN	R/W	0h	Contention LP0 error for lane #4 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
25	RESERVED	R-0	0h	Write 0's for future compatibility. Reads returns 0.
25	ERRCONTENTIONLP1_3_IRQ_EN	R/W	0h	Contention LP1 error for lane #3 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
24	RESERVED	R-0	0h	
24	ERRCONTENTIONLP0_3_IRQ_EN	R/W	0h	Contention LP0 error for lane #3 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
23	ERRCONTENTIONLP1_2_IRQ_EN	R/W	0h	Contention LP1 error for lane #2 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
22	ERRCONTENTIONLP0_2_IRQ_EN	R/W	0h	Contention LP0 error for lane #2 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
21	ERRCONTENTIONLP1_1_IRQ_EN	R/W	0h	Contention LP1 error for lane #1 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
20	ERRCONTENTIONLP0_1_IRQ_EN	R/W	0h	Contention LP0 error for lane #1 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
19	RESERVED	R-0	0h	
19	STATEULPS5_IRQ_EN	R/W	0h	Lane #5 in Ultra Low Power State 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
18	RESERVED	R-0	0h	
18	STATEULPS4_IRQ_EN	R/W	0h	Lane #4 in Ultra Low Power State 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
17	RESERVED	R-0	0h	
17	STATEULPS3_IRQ_EN	R/W	0h	Lane #3 in Ultra Low Power State 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
16	STATEULPS2_IRQ_EN	R/W	0h	Lane #2 in Ultra Low Power State 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
15	STATEULPS1_IRQ_EN	R/W	0h	Lane #1 in Ultra Low Power State 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
14	RESERVED	R-0	0h	
14	ERRCONTROL5_IRQ_EN	R/W	0h	Control error for lane #5 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
13	RESERVED	R-0	0h	

Table 14-164. CSI2_COMPLEXIO_IRQENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	ERRCONTROL4_IRQ_EN	R/W	0h	Control error for lane #4 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
12	RESERVED	R-0	0h	
12	ERRCONTROL3_IRQ_EN	R/W	0h	Control error for lane #3 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
11	ERRCONTROL2_IRQ_EN	R/W	0h	Control error for lane #2 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
10	ERRCONTROL1_IRQ_EN	R/W	0h	Control error for lane #1 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
9	RESERVED	R-0	0h	
9	ERRESC5_IRQ_EN	R/W	0h	Escape entry error for lane #5 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
8	RESERVED	R-0	0h	
8	ERRESC4_IRQ_EN	R/W	0h	Escape entry error for lane #4 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
7	RESERVED	R-0	0h	
7	ERRESC3_IRQ_EN	R/W	0h	Escape entry error for lane #3 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
6	ERRESC2_IRQ_EN	R/W	0h	Escape entry error for lane #2 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
5	ERRESC1_IRQ_EN	R/W	0h	Escape entry error for lane #1 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
4	RESERVED	R-0	0h	
4	ERRSYNCSESC5_IRQ_EN	R/W	0h	Low power Data transmission synchronization error for lane #5 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
3	RESERVED	R-0	0h	
3	ERRSYNCSESC4_IRQ_EN	R/W	0h	Low power Data transmission synchronization error for lane #4 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
2	RESERVED	R-0	0h	
2	ERRSYNCSESC3_IRQ_EN	R/W	0h	Low power Data transmission synchronization error for lane #3 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
1	ERRSYNCSESC2_IRQ_EN	R/W	0h	Low power Data transmission synchronization error for lane #2 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
0	ERRSYNCSESC1_IRQ_EN	R/W	0h	Low power Data transmission synchronization error for lane #1 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs

14.4.4.11 CSI2_CLK_CTRL Register (Offset = 54h) [reset = 1h]

CSI2_CLK_CTRL is shown in Figure 14-169 and described in Table 14-165.

Return to Summary Table.

CLOCK CONTROL This register controls the CLOCK GENERATION. The register can be modified only when IF_EN is reset.

Figure 14-169. CSI2_CLK_CTRL Register

31	30	29	28	27	26	25	24
PLL_PWR_CMD		PLL_PWR_STATUS		RESERVED			
R/W-0h		R-0h		R-0-0h			
23	22	21	20	19	18	17	16
RESERVED		LP_RX_SYNCHRO_ENABLE	LP_CLK_ENABLE	HS_MANUAL_STOP_CTRL	HS_AUTO_STOP_ENABLE	LP_CLK_NULL_PACKET_SIZE	
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
LP_CLK_NULL_PACKET_ENABLE	CIO_CLK_ICG	DDR_CLK_ALWAYS_ON	LP_CLK_DIVISOR				
R/W-0h	R/W-0h	R/W-0h	R/W-1h				
7	6	5	4	3	2	1	0
LP_CLK_DIVISOR							
R/W-1h							

Table 14-165. CSI2_CLK_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	PLL_PWR_CMD	R/W	0h	Command for power control of the CSI2 PLL Control module 0h (R/W) = Command to change to OFF state 1h (R/W) = Command to change to ON state for PLL only (HSDIVISER is OFF) 2h (R/W) = Command to change to ON state for both PLL and HSDIVISER 3h (R/W) = Command to change to ON state for both PLL and HSDIVISER (no clock output to the CSI2 complex IO)
29-28	PLL_PWR_STATUS	R	0h	Status of the power control of the CSI2 PLL Control module 0h (R) = CSI2 PLL Control module in OFF state 1h (R) = CSI2 PLL Control module in ON state for PLL only (HSDIVISER is OFF) 2h (R) = CSI2 PLL Control module in ON state for both PLL and HSDIVISER 3h (R) = CSI2 PLL Control module in ON state for both PLL and HSDIVISER (no clock output to the CSI2 complex IO)
27-22	RESERVED	R-0	0h	
21	LP_RX_SYNCHRO_ENABLE	R/W	0h	Defines if the functional is higher or lower than 30 MHz. The information is used to define synchronization to be used for RxValidEsc. 0h (R/W) = The CSI2 functional clock is equal or slower than 30 MHz. The synchronization is falling/rising. 1h (R/W) = The CSI2 functional clock is higher than 30 MHz. The synchronization is rising/rising.
20	LP_CLK_ENABLE	R/W	0h	Controls the gating of the TXCLKESC clock. 0h (R/W) = Disabled. The clock is not generated. The value of LP_CLK_DIVISOR is not used and does not need to be programmed. 1h (R/W) = Enabled. The clock is generated. The value of LP_CLK_DIVISOR is used and needs to be programmed.

Table 14-165. CSI2_CLK_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HS_MANUAL_STOP_CTRL	R/W	0h	In case HS_AUTO_STOP_ENABLE=0, the bit-field allows manual control of the assertion/de-assertion of the signal CSI2StopClk by the user. 0h (R/W) = CSI2StopClk de-assertion unconditionally. 1h (R/W) = CSI2StopClk assertion unconditionally.
18	HS_AUTO_STOP_ENABLE	R/W	0h	Enables the automatic assertion/de-assertion of CSI2StopClk signal. 0h (R/W) = Auto mode disabled. 1h (R/W) = Auto mode enabled.
17-16	LP_CLK_NULL_PACKET_SIZE	R/W	0h	Indicates the size of LP NULL Packets to be sent automatically when after the last LP packet transfer. It is used by the receiver to drain its internal pipeline. The valid values are from 0 to 3 bytes for the payload size.
15	LP_CLK_NULL_PACKET_ENABLE	R/W	0h	Enables the generation of NULL packet in low speed. 0h (R/W) = Disabled. The NULL packet is not sent in LP mode after the last LP packet. 1h (R/W) = Enabled. The NULL packet is sent in LP mode after the last LP packet.
14	CIO_CLK_ICG	R/W	0h	Gates SCPClk clock provided to CSI2-PHY and PLL-CTRL module. 0h (R/W) = Disabled. SCPClk is not generated. It remains at 0. 1h (R/W) = Enabled. SCPClk is generated (OCP_CLK/4)
13	DDR_CLK_ALWAYS_ON	R/W	0h	Defines if the DDR clock is also sent when there is no HS packets sent to the peripheral (low power mode). So TXRequest for the clock lane is not de-asserted. 0h (R/W) = Disabled. The DDR clock is only provided when HS packets are sent. 1h (R/W) = Enabled. The DDR clock is always sent to the peripheral regardless of the state of the data lanes (HS or LS mode).
12-0	LP_CLK_DIVISOR	R/W	1h	Defines the ratio to be used for the generation of the Low Power mode clock from CSI2 functional clock. The supported values are from 1 to 8191(the value 0 is invalid). The output frequency shall be in the range between 20 MHz and 32 kHz.

14.4.4.12 CSI2_TIMING1 Register (Offset = 58h) [reset = 7FFF7FFFh]

CSI2_TIMING1 is shown in Figure 14-170 and described in Table 14-166.

Return to Summary Table.

TIMING1 REGISTER This register controls the CSI2 Protocol Engine module timers. Any bit-field can be modified while CSI2_CTRL.IF_EN is set to '1'. It is used to indicate the number of CSI2_CLK functional clock cycles for the timers FORCE_TX_STOP_TIMER and TA_TO_TIMER

Figure 14-170. CSI2_TIMING1 Register

31	30	29	28	27	26	25	24
TA_TO	TA_TO_X16	TA_TO_X8	TA_TO_COUNTER				
R/W-0h	R/W-1h	R/W-1h	R/W-1FFFh				
23	22	21	20	19	18	17	16
TA_TO_COUNTER							
R/W-1FFFh							
15	14	13	12	11	10	9	8
FORCE_TX_STOP_MODE_IO	STOP_STATE_X16_IO	STOP_STATE_X4_IO	STOP_STATE_COUNTER_IO				
R/W-0h	R/W-1h	R/W-1h	R/W-1FFFh				
7	6	5	4	3	2	1	0
STOP_STATE_COUNTER_IO							
R/W-1FFFh							

Table 14-166. CSI2_TIMING1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TA_TO	R/W	0h	Enables the turn-around timer 0h (R/W) = Turn-around counter is disabled. 1h (R/W) = Turn-around counter is enabled (required to receive TA interrupt in case the turn-around procedure is not successful).
30	TA_TO_X16	R/W	1h	Multiplication factor for the number of CSI2_CLK functional clock cycles defined in TA_TO_COUNTER bit-field 0h (R/W) = The number of CSI2_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 1x 1h (R/W) = The number of CSI2_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 16x
29	TA_TO_X8	R/W	1h	Multiplication factor for the number of CSI2_CLK functional clock cycles defined in TA_TO_COUNTER bit-field 0h (R/W) = The number of CSI2_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 1x 1h (R/W) = The number of CSI2_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 8x
28-16	TA_TO_COUNTER	R/W	1FFFh	Turn around counter. It indicates the number of CSI2_CLK function clock to wait for the change of the Direction PPI signal according to the TurnRequest signal The value is from 0 to 8191.
15	FORCE_TX_STOP_MODE_IO	R/W	0h	Control of ForceTxStopMode signal 0h (R/W) = De-assertion of ForceTxStopMode. The HW reset the bit at the end of the ForceTXStopMode assertion. The SW can reset the bit in order to stop the assertion of the ForceTXStopMode signal prior to the completion of the period. 1h (R/W) = Assertion of ForceTxStopMode

Table 14-166. CSI2_TIMING1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	STOP_STATE_X16_IO	R/W	1h	Multiplication factor for the number of CSI2_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO bit-field 0h (R/W) = The number of CSI2_CLK functional clock cycles defined in STOP_STATE _COUNTER_IO is multiplied by 1x 1h (R/W) = The number of CSI2_CLK functional clock cycles defined in STOP_STATE _COUNTER_IO is multiplied by 16x
13	STOP_STATE_X4_IO	R/W	1h	Multiplication factor for the number of CSI2_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO bit-field 0h (R/W) = The number of CSI2_CLK functional clock cycles defined in STOP_STATE _COUNTER is multiplied by 1x 1h (R/W) = The number of CSI2_CLK functional clock cycles defined in STOP_STATE _COUNTER_IO is multiplied by 4x
12-0	STOP_STATE_COUNTER_IO	R/W	1FFFh	Stop state counter. It indicates the number of CSI2_CLK function clock to assert ForceTXStopMode signal. The value is from 0 to 8191.

14.4.4.13 CSI2_TIMING2 Register (Offset = 5Ch) [reset = 7FFF7FFFh]

CSI2_TIMING2 is shown in Figure 14-171 and described in Table 14-167.

Return to Summary Table.

TIMING2 REGISTER This register controls the CSI2 Protocol Engine module timers. Any bit-field can be modified while CSI2_CTRL.IF_EN is set to '1'. It is used to indicate the number of CSI2_CLK functional clock cycles for the timers HS_TX_TIMER and LP_RX_TIMER

Figure 14-171. CSI2_TIMING2 Register

31	30	29	28	27	26	25	24
HS_TX_TO	HS_TX_TO_X64	HS_TX_TO_X16	HS_TX_TO_COUNTER				
R/W-0h	R/W-1h	R/W-1h	R/W-1FFFh				
23	22	21	20	19	18	17	16
HS_TX_TO_COUNTER							
R/W-1FFFh							
15	14	13	12	11	10	9	8
LP_RX_TO	LP_RX_TO_X16	LP_RX_TO_X4	LP_RX_TO_COUNTER				
R/W-0h	R/W-1h	R/W-1h	R/W-1FFFh				
7	6	5	4	3	2	1	0
LP_RX_TO_COUNTER							
R/W-1FFFh							

Table 14-167. CSI2_TIMING2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HS_TX_TO	R/W	0h	Enables the HS TX timer. 0h (R/W) = Time-out counter is disabled. 1h (R/W) = Time-out counter is enabled (required to receive TA interrupt in case the turn-around procedure is not successful).
30	HS_TX_TO_X64	R/W	1h	Multiplication factor for the number of BYTE_CLK functional clock cycles defined in HS_TX_COUNTER bit-field 0h (R/W) = The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 1x 1h (R/W) = The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 64x
29	HS_TX_TO_X16	R/W	1h	Multiplication factor for the number of BYTE_CLK functional clock cycles defined in HS_TX_COUNTER bit 0h (R/W) = The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 1x 1h (R/W) = The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 16x
28-16	HS_TX_TO_COUNTER	R/W	1FFFh	HS_TX_TIMER counter. It indicates the number of BYTE_CLK function clock for the HS TX timer. The value is from 0 to 8191.
15	LP_RX_TO	R/W	0h	Enables the LP RX timer. 0h (R/W) = Turn-around counter is disabled. 1h (R/W) = Turn-around counter is enabled (required to receive TA interrupt in case the turn-around procedure is not successful).
14	LP_RX_TO_X16	R/W	1h	Multiplication factor for the number of CSI2_CLK functional clock cycles defined in LP_RX_COUNTER bit-field 0h (R/W) = The number of CSI2_CLK functional clock cycles defined in LP_RX_TO_COUNTER is multiplied by 1x 1h (R/W) = The number of CSI2_CLK functional clock cycles defined in LP_RX_TO_COUNTER is multiplied by 16x

Table 14-167. CSI2_TIMING2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	LP_RX_TO_X4	R/W	1h	Multiplication factor for the number of CSI2_CLK functional clock cycles defined in LP_RX_COUNTER bit 0h (R/W) = The number of CSI2_CLK functional clock cycles defined in LP_RX_TO_COUNTER is multiplied by 1x 1h (R/W) = The number of CSI2_CLK functional clock cycles defined in LP_RX_TO_COUNTER is multiplied by 4x
12-0	LP_RX_TO_COUNTER	R/W	1FFFh	LP_RX_TIMER counter. It indicates the number of CSI2_CLK function clock for the LP RX timer. The value is from 0 to 8191.

14.4.4.14 CSI2_VM_TIMING1 Register (Offset = 60h) [reset = 0h]

CSI2_VM_TIMING1 is shown in Figure 14-172 and described in Table 14-168.

Return to Summary Table.

VIDEO MODE TIMING REGISTER This register defines the video mode timing.

Figure 14-172. CSI2_VM_TIMING1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSA								HFP												HBP											
R/W-0h								R/W-0h												R/W-0h											

Table 14-168. CSI2_VM_TIMING1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	HSA	R/W	0h	Defines the horizontal Sync active period used in video mode in number of byte clock cycles (PPI clock) The supported values are from 0 to 255.
23-12	HFP	R/W	0h	Defines the horizontal front porch used in video mode in number of byte clock cycles (PPI clock) The supported values are from 0 to 4095
11-0	HBP	R/W	0h	Defines the horizontal back porch used in video mode in number of byte clock cycles (PPI clock) The supported values are from 0 to 4095

14.4.4.15 CSI2_VM_TIMING2 Register (Offset = 64h) [reset = 0h]

CSI2_VM_TIMING2 is shown in Figure 14-173 and described in Table 14-169.

Return to Summary Table.

VIDEO MODE TIMING REGISTER This register defines the video mode timing.

Figure 14-173. CSI2_VM_TIMING2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				WINDOW_SYNC				VSA							
R-0-0h				R/W-0h				R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VFP								VBP							
R/W-0h								R/W-0h							

Table 14-169. CSI2_VM_TIMING2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	
27-24	WINDOW_SYNC	R/W	0h	Number of BYTE clock cycles for the synchronization window. An interrupt for synchronization lost is generated when the received synchornization on video port is not inside the window. CSI2 does not change its own timings if the synch is inside the window. The valid values are from 4 to 15.
23-16	VSA	R/W	0h	Defines the vertical Sync active period used in video mode in number of lines. The supported values are from 0 to 255 It is used to generate the short packet for End of Vertical synchronization.
15-8	VFP	R/W	0h	Defines the vertical front porch used in video mode in number of lines. The supported values are from 0 to 255
7-0	VBP	R/W	0h	Defines the vertical back porch used in video mode in number of lines. The supported values are from 0 to 255

14.4.4.16 CSI2_VM_TIMING3 Register (Offset = 68h) [reset = 0h]

CSI2_VM_TIMING3 is shown in Figure 14-174 and described in Table 14-170.

Return to Summary Table.

VIDEO MODE TIMING REGISTER This register defines the video mode timing.

Figure 14-174. CSI2_VM_TIMING3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TL																VACT															
R/W-0h																R/W-0h															

Table 14-170. CSI2_VM_TIMING3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TL	R/W	0h	Defines the number of length of the line in video mode in number of byte clock cycles (PPI clock) The supported values are from 0 to 8192. The values from 8193 to 65535 are not supported.
15-0	VACT	R/W	0h	Defines the number of active lines used in video mode. The supported values are from 0 to 65535

14.4.4.17 CSI2_CLK_TIMING Register (Offset = 6Ch) [reset = 101h]

CSI2_CLK_TIMING is shown in Figure 14-175 and described in Table 14-171.

Return to Summary Table.

CLOCK TIMING REGISTER This register controls the CSI2 Protocol Engine module timers. This register shall not be modified while CSI2_CTRL.IF_EN is set to '1'.

Figure 14-175. CSI2_CLK_TIMING Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDR_CLK_PRE								DDR_CLK_POST							
R/W-1h								R/W-1h							

Table 14-171. CSI2_CLK_TIMING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	
15-8	DDR_CLK_PRE	R/W	1h	Indicates the number of PPI Byte clock cycles between the start of the DDR clock and the assertion of the data request signal. The values from 1 to 255 are valid. The value 0 is reserved. The value is not used if CSI2_CLK_CTRL.DDR_CLK_ALWAYS_ON is set to '1' since the DDR clock is always present.
7-0	DDR_CLK_POST	R/W	1h	Indicates the number of PPI Byte clock cycles after the de-assertion of the data request signal and the stop of the DDR clock. The values from 1 to 255 are valid. The value 0 is reserved. The value is not used if CSI2_CLK_CTRL.DDR_CLK_ALWAYS_ON is set to '1' since the DDR clock is always present.

14.4.4.18 CSI2_TX_FIFO_VC_SIZE Register (Offset = 70h) [reset = 0h]

CSI2_TX_FIFO_VC_SIZE is shown in Figure 14-176 and described in Table 14-172.

Return to Summary Table.

Defines the corresponding memory entries allocated for each virtual channel. The virtual channel shall be disabled in order to allocate/un-allocate some entries in the TX FIFO.

Figure 14-176. CSI2_TX_FIFO_VC_SIZE Register

31	30	29	28	27	26	25	24
VC3_FIFO_SIZE				RESERVED	VC3_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		
23	22	21	20	19	18	17	16
VC2_FIFO_SIZE				RESERVED	VC2_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		
15	14	13	12	11	10	9	8
VC1_FIFO_SIZE				RESERVED	VC1_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		
7	6	5	4	3	2	1	0
VC0_FIFO_SIZE				RESERVED	VC0_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		

Table 14-172. CSI2_TX_FIFO_VC_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	VC3_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 3.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
27	RESERVED	R-0	0h	
26-24	VC3_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 3.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.
23-20	VC2_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 2.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
19	RESERVED	R-0	0h	
18-16	VC2_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 2.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.
15-12	VC1_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 1.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
11	RESERVED	R-0	0h	
10-8	VC1_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 1.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.
7-4	VC0_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 0.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
3	RESERVED	R-0	0h	
2-0	VC0_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 0.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.

14.4.4.19 CSI2_RX_FIFO_VC_SIZE Register (Offset = 74h) [reset = 0h]

CSI2_RX_FIFO_VC_SIZE is shown in Figure 14-177 and described in Table 14-173.

Return to Summary Table.

Defines the corresponding memory entries allocated for each virtual channel and the addresses. The virtual channel shall be disabled in order to allocate/un-allocate some entries in the RX FIFO.

Figure 14-177. CSI2_RX_FIFO_VC_SIZE Register

31	30	29	28	27	26	25	24
VC3_FIFO_SIZE				RESERVED	VC3_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		
23	22	21	20	19	18	17	16
VC2_FIFO_SIZE				RESERVED	VC2_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		
15	14	13	12	11	10	9	8
VC1_FIFO_SIZE				RESERVED	VC1_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		
7	6	5	4	3	2	1	0
VC0_FIFO_SIZE				RESERVED	VC0_FIFO_ADD		
R/W-0h				R-0-0h	R/W-0h		

Table 14-173. CSI2_RX_FIFO_VC_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	VC3_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 3.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
27	RESERVED	R-0	0h	
26-24	VC3_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 3.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.
23-20	VC2_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 2.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
19	RESERVED	R-0	0h	
18-16	VC2_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 2.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.
15-12	VC1_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 1.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
11	RESERVED	R-0	0h	
10-8	VC1_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 1.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.
7-4	VC0_FIFO_SIZE	R/W	0h	Size of the FIFO allocated for virtual channel 0.The valid values are from 0 to 8 for a size of the FIFO of 256x33bits corresponding to 0x33bits, 32x33bits, 64x33bits...
3	RESERVED	R-0	0h	
2-0	VC0_FIFO_ADD	R/W	0h	Address of the space allocated in the FIFO for virtual channel 0.The valid values are from 0 to 7 for a size of the FIFO of 256x33bits corresponding to 0, 32, 64,... for the entry address.

14.4.4.20 CSI2_COMPLEXIO_CFG2 Register (Offset = 78h) [reset = 0h]

CSI2_COMPLEXIO_CFG2 is shown in Figure 14-178 and described in Table 14-174.

Return to Summary Table.

COMPLEXIO CONFIGURATION REGISTER for the complex IO This register contains the lane configuration for the ULPS for each lane.

Figure 14-178. CSI2_COMPLEXIO_CFG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						LP_BUSY	HS_BUSY
R-0-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED						LANE5_ULPS_SIG2	RESERVED
R-0-0h						R/W-0h	R-0-0h
7	6	5	4	3	2	1	0
LANE3_ULPS_SIG2	LANE2_ULPS_SIG2	LANE1_ULPS_SIG2	LANE5_ULPS_SIG1	LANE4_ULPS_SIG1	LANE3_ULPS_SIG1	LANE2_ULPS_SIG1	LANE1_ULPS_SIG1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-174. CSI2_COMPLEXIO_CFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	
17	LP_BUSY	R	0h	Indicates when there are still pending operations for VCs configured for LP mode. Forced to 1 when at least one VC is enabled and configured for LP mode. 0h (R) = LP logic is idle 1h (R) = LP logic is active
16	HS_BUSY	R	0h	Indicates when there are still pending operations for VCs configured for HS mode. Forced to 1 when at least one VC is enabled and configured for HS mode. 0h (R) = HS logic is idle 1h (R) = HS logic is active
15-10	RESERVED	R-0	0h	
9	RESERVED	R-0	0h	
9	LANE5_ULPS_SIG2	R/W	0h	Enables the ULPS for the lane #5. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane #5 is a data lane. The state of the signal TxUlpsClk is changed if lane #5 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ:Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc.

Table 14-174. CSI2_COMPLEXIO_CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	LANE4_ULPS_SIG2	R/W	0h	Enables the ULPS for the lane #4. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane #4 is a data lane. The state of the signal TxUlpsClk is changed if lane #4 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ: Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc.
8	RESERVED	R-0	0h	
7	RESERVED	R-0	0h	
7	LANE3_ULPS_SIG2	R/W	0h	Enables the ULPS for the lane #3. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane #3 is a data lane. The state of the signal TxUlpsClk is changed if lane #3 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ: Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc.
6	LANE2_ULPS_SIG2	R/W	0h	Enables the ULPS for the lane #2. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane #2 is a data lane. The state of the signal TxUlpsClk is changed if lane #2 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ: Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc.

Table 14-174. CSI2_COMPLEXIO_CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	LANE1_ULPS_SIG2	R/W	0h	Enables the ULPS for the lane #1. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane #1 is a data lane. The state of the signal TxUlpsClk is changed if lane #1 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ: Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc.
4	RESERVED	R-0	0h	Write 0's for future compatibility. Reads returns 0.
4	LANE5_ULPS_SIG1	R/W	0h	Enables the ULPS for the lane #5. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane #5 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ:Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc
3	RESERVED	R-0	0h	
3	LANE4_ULPS_SIG1	R/W	0h	Enables the ULPS for the lane #4. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane #4 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ: Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc
2	RESERVED	R-0	0h	

Table 14-174. CSI2_COMPLEXIO_CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	LANE3_ULPS_SIG1	R/W	0h	Enables the ULPS for the lane #3. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane #3 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ: Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc
1	LANE2_ULPS_SIG1	R/W	0h	Enables the ULPS for the lane #2. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane #2 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ: Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc
0	LANE1_ULPS_SIG1	R/W	0h	Enables the ULPS for the lane #1. The HW shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the CSI2 protocol engine and the CSI2 protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane #1 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0h (R/W) = READ:Inactive state effective. WRITE: Request to change to inactive state 1h (R/W) = READ:Active state effective WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkEsc

14.4.4.21 CSI2_RX_FIFO_VC_FULLNESS Register (Offset = 7Ch) [reset = 0h]

CSI2_RX_FIFO_VC_FULLNESS is shown in Figure 14-179 and described in Table 14-175.

Return to Summary Table.

Defines the fullness of each space allocated for each virtual channel.

Figure 14-179. CSI2_RX_FIFO_VC_FULLNESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VC3_FIFO_FULLNESS								VC2_FIFO_FULLNESS							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC1_FIFO_FULLNESS								VC0_FIFO_FULLNESS							
R-0h								R-0h							

Table 14-175. CSI2_RX_FIFO_VC_FULLNESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VC3_FIFO_FULLNESS	R	0h	Fullness of the FIFO allocated for virtual channel 3.The valid values are from 0 to CSI2_GNQ.RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.RX_FIFODEPTH x33-bit.
23-16	VC2_FIFO_FULLNESS	R	0h	Fullness of the FIFO allocated for virtual channel 2.The valid values are from 0 to CSI2_GNQ.RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.RX_FIFODEPTH x33-bit.
15-8	VC1_FIFO_FULLNESS	R	0h	Fullness of the FIFO allocated for virtual channel 1.The valid values are from 0 to CSI2_GNQ.RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.RX_FIFODEPTH x33-bit.
7-0	VC0_FIFO_FULLNESS	R	0h	Fullness of the FIFO allocated for virtual channel 0.The valid values are from 0 to CSI2_GNQ.RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.RX_FIFODEPTH x33-bit.

14.4.4.22 CSI2_VM_TIMING4 Register (Offset = 80h) [reset = 0h]

CSI2_VM_TIMING4 is shown in Figure 14-180 and described in Table 14-176.

Return to Summary Table.

VIDEO MODE TIMING REGISTER This register defines the video mode timing.

Figure 14-180. CSI2_VM_TIMING4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								HSA_HS_INTERLEAVING							
R-0-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HFP_HS_INTERLEAVING								HBP_HS_INTERLEAVING							
R/W-0h								R/W-0h							

Table 14-176. CSI2_VM_TIMING4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	
23-16	HSA_HS_INTERLEAVING	R/W	0h	Defines the number of HS byte clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during HSA blanking period. The supported values are from 0 to 255.
15-8	HFP_HS_INTERLEAVING	R/W	0h	Defines the number of HS byte clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during HFP blanking period. The supported values are from 0 to 255
7-0	HBP_HS_INTERLEAVING	R/W	0h	Defines the number of HS byte clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during HBP blanking period. The supported values are from 0 to 255

14.4.4.23 CSI2_TX_FIFO_VC_EMPTINESS Register (Offset = 84h) [reset = 0h]

CSI2_TX_FIFO_VC_EMPTINESS is shown in Figure 14-181 and described in Table 14-177.

Return to Summary Table.

Defines the emptiness of each space allocated for each virtual channel.

Figure 14-181. CSI2_TX_FIFO_VC_EMPTINESS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VC3_FIFO_EMPTINESS								VC2_FIFO_EMPTINESS							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC1_FIFO_EMPTINESS								VC0_FIFO_EMPTINESS							
R-0h								R-0h							

Table 14-177. CSI2_TX_FIFO_VC_EMPTINESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VC3_FIFO_EMPTINESS	R	0h	Emptiness of the FIFO allocated for virtual channel 3.The valid values are from 0 to CSI2_GNQ.TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.TX_FIFODEPTH x33-bit.
23-16	VC2_FIFO_EMPTINESS	R	0h	Emptiness of the FIFO allocated for virtual channel 2.The valid values are from 0 to CSI2_GNQ.TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.TX_FIFODEPTH x33-bit.
15-8	VC1_FIFO_EMPTINESS	R	0h	Emptiness of the FIFO allocated for virtual channel 1.The valid values are from 0 to CSI2_GNQ.TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.TX_FIFODEPTH x33-bit.
7-0	VC0_FIFO_EMPTINESS	R	0h	Emptiness of the FIFO allocated for virtual channel 0.The valid values are from 0 to CSI2_GNQ.TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to CSI2_GNQ.TX_FIFODEPTH x33-bit.

14.4.4.24 CSI2_VM_TIMING5 Register (Offset = 88h) [reset = 0h]

CSI2_VM_TIMING5 is shown in Figure 14-182 and described in Table 14-178.

Return to Summary Table.

VIDEO MODE TIMING REGISTER This register defines the video mode timing.

Figure 14-182. CSI2_VM_TIMING5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								HSA_LP_INTERLEAVING							
R-0-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HFP_LP_INTERLEAVING								HBP_LP_INTERLEAVING							
R/W-0h								R/W-0h							

Table 14-178. CSI2_VM_TIMING5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	
23-16	HSA_LP_INTERLEAVING	R/W	0h	Defines the number of bytes of Low Power command mode packets that can be sent on PPI link during HSA blanking period. The supported values are from 0 to 255.
15-8	HFP_LP_INTERLEAVING	R/W	0h	Defines the number of bytes of Low Power command mode packets that can be sent on PPI link during HFP blanking period. The supported values are from 0 to 255
7-0	HBP_LP_INTERLEAVING	R/W	0h	Defines the number of bytes of Low Power command mode packets that can be sent on PPI link during HBP blanking period. The supported values are from 0 to 255

14.4.4.25 CSI2_VM_TIMING6 Register (Offset = 8Ch) [reset = 0h]

CSI2_VM_TIMING6 is shown in Figure 14-183 and described in Table 14-179.

Return to Summary Table.

VIDEO MODE TIMING REGISTER This register defines the video mode timing.

Figure 14-183. CSI2_VM_TIMING6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BL_HS_INTERLEAVING															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BL_LP_INTERLEAVING															
R/W-0h															

Table 14-179. CSI2_VM_TIMING6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BL_HS_INTERLEAVING	R/W	0h	Defines the number of HS byte clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during blanking periods during VSA, VBP, VFP periods inside one video frame on PPI link. The supported values are from 0 to 65535 .
15-0	BL_LP_INTERLEAVING	R/W	0h	Defines the maximum number of bytes of Low Power command mode packets that can be sent on PPI link during blanking periods during VSA, VBP or VFP periods inside one video frame on PPI link. The supported values are from 0 to 65535

14.4.4.26 CSI2_VM_TIMING7 Register (Offset = 90h) [reset = 0h]

CSI2_VM_TIMING7 is shown in Figure 14-184 and described in Table 14-180.

Return to Summary Table.

Defines the minimum number of HS bytes clock cycles that are required to allow for the delays in entering and exiting HS mode. The supported values are from 0 to 65535

Figure 14-184. CSI2_VM_TIMING7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENTER_HS_MODE_LATENCY															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXIT_HS_MODE_LATENCY															
R/W-0h															

Table 14-180. CSI2_VM_TIMING7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ENTER_HS_MODE_LATENCY	R/W	0h	Defines the number of HS byte clock cycles necessary for entering to HS mode. It corresponds to the delay in number of HS clock cycles from assertion of TxRequestHS signal to 1 until assertion of TxReadyHS signal to 1. The supported values are from 0 to 65535 .
15-0	EXIT_HS_MODE_LATENCY	R/W	0h	Defines the number of HS byte clock cycles necessary for exiting from HS mode. It corresponds to the maximum delay in number of HS byte clock from de-assertion of TxRequestHS signal until PPI link is in LP-11 state from which a new entrance to HS mode can be initiated which does not take more than ENTER_HS_MODE_LATENCY clock cycles. The supported values are from 0 to 65535

14.4.4.27 CSI2_STOPCLK_TIMING Register (Offset = 94h) [reset = 80h]

CSI2_STOPCLK_TIMING is shown in Figure 14-185 and described in Table 14-181.

Return to Summary Table.

Number of functional clock cycles to wait for TxByteClkHS to stop/start after change in CSI2StopClk signal

Figure 14-185. CSI2_STOPCLK_TIMING Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSI2_STOPCLK_LATENCY							
R-0-0h								R/W-80h							

Table 14-181. CSI2_STOPCLK_TIMING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	
7-0	CSI2_STOPCLK_LATENCY	R/W	80h	Clock gating latency from CSI2 Protocol to TxByteClkHS

14.4.4.28 CSI2_CTRL2 Register (Offset = 98h) [reset = 100h]

CSI2_CTRL2 is shown in Figure 14-186 and described in Table 14-182.

Return to Summary Table.

Additional control bits for use with Video Port 2

Figure 14-186. CSI2_CTRL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED		LINE_BUFFER		VP_VSYNC_POL	VP_HSYNC_POL	VP_DE_POL	VP_CLK_POL
R-0-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-1h
7	6	5	4	3	2	1	0
VP_DATA_BUS_WIDTH		RESERVED	VP_CLK_RATIO	RESERVED			
R/W-0h		R-0-0h	R/W-0h	R-0-0h			

Table 14-182. CSI2_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	
13-12	LINE_BUFFER	R/W	0h	Number of line buffers to be used while receiving data on the video port. The valid values are from 0 to CSI2_GNQ.VP2_NB_LINE_BUFFER. 0h (R/W) = No line buffer 1h (R/W) = 1 line buffer 2h (R/W) = 2 line buffers
11	VP_VSYNC_POL	R/W	0h	VP vertical synchronization signal polarity 0h (R/W) = VSYNC signal on the video port is active low. 1h (R/W) = VSYNC signal on the video port is active high.
10	VP_HSYNC_POL	R/W	0h	VP horizontal synchronization signal polarity 0h (R/W) = HSYNC signal on the video port is active low. 1h (R/W) = HSYNC signal on the video port is active high.
9	VP_DE_POL	R/W	0h	VP data enable signal polarity 0h (R/W) = DE signal on the video port is active low. 1h (R/W) = DE signal on the video port is active high.
8	VP_CLK_POL	R/W	1h	VP clock polarity 0h (R/W) = The CSI2 Protocol Engine module captures the data on the VP on the pixel clock falling edge. The module connected to the VP shall drive the data on the pixel clock rising edge. 1h (R/W) = The CSI2 Protocol Engine module captures the data on the VP on the pixel clock raising edge. The module connected to the VP shall drive the data on the pixel clock falling edge.
7-6	VP_DATA_BUS_WIDTH	R/W	0h	Defines the size of the video port data bus 0h (R/W) = 16-bits data width (LSB of the 24-bit video port data bus) 1h (R/W) = 18-bits data width (LSB of the 24-bit video port data bus) 2h (R/W) = 24-bits data width (LSB of the 24-bit video port data bus)
5	RESERVED	R-0	0h	

Table 14-182. CSI2_CTRL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	VP_CLK_RATIO	R/W	0h	The field indicates the clock ratio between VP.CLK and VP.PCLK. The clock VP.PCLK is generated from VP.CLK. It is divided down. The information is only used when the video port is used to provide data in command mode. In the case of video mode, it is not used. 0h (R/W) = The clock VP.PCLK is the clock VP.CLK divided by 2. The duty cycle of VP.PCLK is 50/50. 1h (R/W) = The clock VP.PCLK is the clock VP.CLK divided by 3 or more. The duty cycle of VP.PCLK is not 50/50 for odd ratio numbers (3,5,7,...).
3-0	RESERVED	R-0	0h	

14.4.4.29 CSI2_VM_TIMING8 Register (Offset = 9Ch) [reset = 0h]

CSI2_VM_TIMING8 is shown in Figure 14-187 and described in Table 14-183.

Return to Summary Table.

VIDEO MODE TIMING REGISTER
This register defines the video mode timing.

Figure 14-187. CSI2_VM_TIMING8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														HFPX	
R-0-0h														R/W-0h	

Table 14-183. CSI2_VM_TIMING8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	
1-0	HFPX	R/W	0h	Extension to the HFP register. Additional bits added to MSB.

14.4.4.30 CSI2_TE_HSYNC_WIDTH_0 to CSI2_TE_HSYNC_WIDTH_1 Register (Offset = A0h + [i * 6h], where i = 0 to 1) [reset = 100h]

CSI2_TE_HSYNC_WIDTH_0 to CSI2_TE_HSYNC_WIDTH_1 is shown in Figure 14-188 and described in Table 14-184.

Return to Summary Table.

The register configures the TE HSYNC minimum pulse width for TE0 and TE1 CMOS signals
The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.

Figure 14-188. CSI2_TE_HSYNC_WIDTH_0 to CSI2_TE_HSYNC_WIDTH_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED												MIN_HSYNC_PULSE_WIDTH			
R-0-0h												R/W-1h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_HSYNC_PULSE_WIDTH								RESERVED							
R/W-1h								R-0-0h							

Table 14-184. CSI2_TE_HSYNC_WIDTH_0 to CSI2_TE_HSYNC_WIDTH_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-8	MIN_HSYNC_PULSE_WIDTH	R/W	1h	Programmable min HSYNC pulse width Minimum HSYNC pulse width. Number of CSI2_CLK clock cycles times 256 to determine when HSYNC pulse occurs. The value 0 is invalid.
7-0	RESERVED	R-0	0h	

14.4.4.31 CSI2_TE_VSYNC_WIDTH_0 to CSI2_TE_VSYNC_WIDTH_1 Register (Offset = A4h + [i * 6h], where i = 0 to 1) [reset = 100h]

CSI2_TE_VSYNC_WIDTH_0 to CSI2_TE_VSYNC_WIDTH_1 is shown in Figure 14-189 and described in Table 14-185.

Return to Summary Table.

The register configures the TE VSYNC minimum pulse width for TE0 and TE1 CMOS signals
The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.

Figure 14-189. CSI2_TE_VSYNC_WIDTH_0 to CSI2_TE_VSYNC_WIDTH_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED												MIN_VSYNC_PULSE_WIDTH			
R-0-0h												R/W-1h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIN_VSYNC_PULSE_WIDTH								RESERVED							
R/W-1h								R-0-0h							

Table 14-185. CSI2_TE_VSYNC_WIDTH_0 to CSI2_TE_VSYNC_WIDTH_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	
19-8	MIN_VSYNC_PULSE_WIDTH	R/W	1h	Programmable min VSYNC pulse width Minimum VSYNC pulse width. Number of CSI2_CLK cycles times 256 to determine when VSYNC pulse occurs. The value 0 is invalid. The value shall be greater than MIN_HSYNC_PULSE_WIDTH when CSI2_TE_HSYNC_NUMBER is greather than 0
7-0	RESERVED	R-0	0h	

14.4.4.32 CSI2_TE_HSYNC_NUMBER_0 to CSI2_TE_HSYNC_NUMBER_1 Register (Offset = A8h + [i * 6h], where i = 0 to 1) [reset = 0h]

CSI2_TE_HSYNC_NUMBER_0 to CSI2_TE_HSYNC_NUMBER_1 is shown in Figure 14-190 and described in Table 14-186.

Return to Summary Table.

The register configures the number of HSYNC to synchronize the beginning of the transfer on CSI2 link based on the number of HSYNC pulse received on the TE line.
The input TE signal is asynchronous and needs to be re-synchronizred to CSI2_CLK clock domain.

Figure 14-190. CSI2_TE_HSYNC_NUMBER_0 to CSI2_TE_HSYNC_NUMBER_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LINE_NUMBER										
R-0-0h																					R/W-0h										

Table 14-186. CSI2_TE_HSYNC_NUMBER_0 to CSI2_TE_HSYNC_NUMBER_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	
10-0	LINE_NUMBER	R/W	0h	Programmable line number Line number from 0 to 2047. Number of HSYNC after the VSYNC occurs before the beginning of the transfer. Any HSYNC before VSYNC is ignored.

14.4.4.33 CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register (Offset = 100h + [i * 8h], where i = 0 to 3) [reset = 0h]

CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 is shown in Figure 14-191 and described in Table 14-187.

Return to Summary Table.

CONTROL REGISTER - Virtual channel This register controls the virtual channel.

Figure 14-191. CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register

31	30	29	28	27	26	25	24
DCS_CMD_CODE	DCS_CMD_ENABLE	DMA_RX_REQ_NB			DMA_RX_THRESHOLD		
R/W-0h	R/W-0h	R/W-0h			R/W-0h		
23	22	21	20	19	18	17	16
DMA_TX_REQ_NB			RX_FIFO_NOT_EMPTY	DMA_TX_THRESHOLD			TX_FIFO_FULL
R/W-0h			R-0h	R/W-0h			R-0h
15	14	13	12	11	10	9	8
VC_BUSY	PP_BUSY	VP_SOURCE	RGB565_ORDER	OCP_DATA_BUS_WIDTH		MODE_SPEED	ECC_TX_EN
R-0h	R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CS_TX_EN	BTA_EN	TX_FIFO_NOT_EMPTY	MODE	BTA_LONG_EN	BTA_SHORT_EN	SOURCE	VC_EN
R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-187. CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DCS_CMD_CODE	R/W	0h	DCS command code value to insert between header and video port or OCP slave data when enabled by DCS_CMD_ENABLE (only when TE mechanism is not used otherwise the bit-field DCS_CMD_CODE is ignored by CSI2 protocol engine). 0h (R/W) = DCS write memory continue code is inserted. 1h (R/W) = DCS write memory start code is inserted.
30	DCS_CMD_ENABLE	R/W	0h	Enables automatic insertion of DCS command codes when data is sourced by the video port. 0h (R/W) = DCS command code is NOT inserted when command mode traffic is coming from the Video Port or OCP slave port. 1h (R/W) = DCS command code is inserted automatically when command mode traffic is coming from the Video Port or OCP slave port.
29-27	DMA_RX_REQ_NB	R/W	0h	Selection of the use of the DMA request (associated to the RX FIFO) 0h (R/W) = DMA_req0 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is greater than 0) 1h (R/W) = DMA_req1 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is greater than 1) 2h (R/W) = DMA_req2 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is greater than 2) 3h (R/W) = DMA_req3 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is equal to 3) 4h (R/W) = No DMA req selected

Table 14-187. CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26-24	DMA_RX_THRESHOLD	R/W	0h	Defines the threshold value for the DMA request (associated to the RX FIFO) 0h (R/W) = 1x 32 bits 1h (R/W) = 2 x 32 bits 2h (R/W) = 4 x 32 bits 3h (R/W) = 8 x 32 bits 4h (R/W) = 16 x 32 bits 5h (R/W) = 32 x 32 bits
23-21	DMA_TX_REQ_NB	R/W	0h	Selection of the use of the DMA request (associated to the TX FIFO) 0h (R/W) = DMA_req0 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is greater than 0) 1h (R/W) = DMA_req1 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is greater than 1) 2h (R/W) = DMA_req2 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is greater than 2) 3h (R/W) = DMA_req3 is selected (valid only if CSI2_GNQ.NB_DMA_REQUEST is equal to 3) 4h (R/W) = No DMA req selected
20	RX_FIFO_NOT_EMPTY	R	0h	FIFO status 0h (R) = The RX FIFO is empty (the FIFO does not contain any data for the virtual channel) 1h (R) = The RX FIFO is not empty (the FIFO contains at least one byte for the virtual channel)
19-17	DMA_TX_THRESHOLD	R/W	0h	Defines the threshold value for the DMA request (associated to the TX FIFO) 0h (R/W) = 1x 32 bits 1h (R/W) = 2 x 32 bits 2h (R/W) = 4 x 32 bits 3h (R/W) = 8 x 32 bits 4h (R/W) = 16 x 32 bits 5h (R/W) = 32 x 32 bits
16	TX_FIFO_FULL	R	0h	FIFO status 0h (R) = The TX FIFO is not full (the FIFO can accept at least one more 32-bit valuel) 1h (R) = The TX FIFO is full
15	VC_BUSY	R	0h	Indicates if previously scheduled activities (packets, BTA) are still being processed. Forced to 1 if VC is enabled. Software should check this bit is 0 before changing channel configuration. 0h (R) = No pending operations for this VC 1h (R) = Pending operations for this VC
14	PP_BUSY	R	0h	Ping-pong buffer busy status. 0h (R) = Software is permitted to write a new header for VP command mode traffic. 1h (R) = Software is NOT permitted to write a new header for VP command mode traffic.
13	VP_SOURCE	R/W	0h	Selection between video port 1 and video port 2. If CSI2_GNQ.NB_VIDEO_PORTS=0, the bit-field is ignored since only video port 1 is present. 0h (R/W) = The video port 1 is selected. 1h (R/W) = The video port 2 is selected.
12	RGB565_ORDER	R/W	0h	Byte order for RBG565 0h (R/W) = Byte order as documented in the MIPI DBI-2 spec 1h (R/W) = Byte order as for video mode

Table 14-187. CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	OCP_DATA_BUS_WIDTH	R/W	0h	Defines the size of the OCP data bus 0h (R/W) = 16-bits data width (LSB of the 32-bit OCP port data bus) 1h (R/W) = 24-bits data width (LSB of the 32-bit OCP port data bus) 2h (R/W) = 2x16-bits data width (first pixel on the LSB of the 32-bit OCP port data bus and second pixel on the MSB of the 32-bit OCP port data bus for the same OCP access) 3h (R/W) = 32-bits data width
9	MODE_SPEED	R/W	0h	Selection of the mode. The information is used by HW only if MODE=COMMAND_MODE otherwise it is ignored. 0h (R/W) = Low power mode (CMOS) is used to send short and long packets to the peripheral. 1h (R/W) = High Speed mode (SLVS) is used to send short and long packets to the peripheral.
8	ECC_TX_EN	R/W	0h	Enables the Error Correction Code generation for the transmit header (short and long packets). 0h (R/W) = Disabled 1h (R/W) = Enabled
7	CS_TX_EN	R/W	0h	Enables the checksum generation for the transmit payload (long packet only). 0h (R/W) = Disabled. The value 0x00 is used. 1h (R/W) = Enabled. The Check-sum value is calculated by HW.
6	BTA_EN	R/W	0h	Send the bus turn around to the peripheral. It can be used when the automatic mode is enabled (BTA_SHORT_EN=1 or/and BTA_LONG_EN=1). In that case only one BTA is sent to the peripheral. The manual mode is used to allow the user to define for which packets, the turn around is required for example getting acknowledge from the peripheral. 0h (R/W) = READS: BTA generation is completed. It is reset by HW when it is completed. WRITES: Cancellation of the BTA generation (not guarantee since it could already on going, shall not be used). 1h (R/W) = READS: BTA generation has been requested by user (it could be on going but not completed). WRITES: Request for BTA generation.
5	TX_FIFO_NOT_EMPTY	R	0h	FIFO status 0h (R) = The TX FIFO is empty (the FIFO does not contain any data for the virtual channel) 1h (R) = The TX FIFO is not empty (the FIFO contains at least one byte for the virtual channel)
4	MODE	R/W	0h	Selection of the mode 0h (R/W) = Command mode. 1h (R/W) = Video mode. The bit-fields MODE_SPEED and SOURCE are not used by HW. The bit-field VP_SOURCE is used to select between video port 1 and video port 2 when two video ports are present.
3	BTA_LONG_EN	R/W	0h	Enables the automatic bus turn-around after completion of each long packet transmission. 0h (R/W) = Disabled 1h (R/W) = Enabled
2	BTA_SHORT_EN	R/W	0h	Enables the automatic bus turn-around after completion of each short packet transmission. 0h (R/W) = Disabled 1h (R/W) = Enabled

Table 14-187. CSI2_VC_CTRL_0 to CSI2_VC_CTRL_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SOURCE	R/W	0h	Selection of the source between OCP and the Video port(s) (used by the HW only if MODE=COMMAND MODE (0x0) otherwise it is ignored). The number of available video ports is defined in CSI2_GNQ.NB_VIDEO_PORTS. 0h (R/W) = All the data are provided by the slave port. Any transfer on the video port is ignored for this virtual channel. 1h (R/W) = If MODE=VIDEO_MODE. the data received on the video port 1 or video port 2, selected using VP_SOURCE (pixels and enabled synchronization events using CSI2_CTRL1.VP_HSYNC_START, CSI2_CTRL1.VP_HSYNC_END, CSI2_CTRL1.VP_VSYNC_START, CSI2_CTRL1.VP_VSYNC_END for video port 1 and CSI2_CTRL2.VP_HSYNC_START, CSI2_CTRL2.VP_HSYNC_END, CSI2_CTRL2.VP_VSYNC_START, CSI2_CTRL2.VP_VSYNC_END for video port 2) are sent on the virtual channel (only one virtual channel can be associated with the video port, it is the SW responsibility to ensure that no more than one virtual channel is enabled with the video port as the main source for data). If MODE=COMMAND_MODE, the VP.STALL signal is used by the protocol engine to indicate when new data are required. The synchronization signals are not generated by the display controller. Regardless of the MODE, no data can be provided on the OCP slave port.
0	VC_EN	R/W	0h	Enables the virtual channel. 0h (R/W) = Disabled. The virtual channel shall be disabled for any register change in the CSI2_VC_... registers the corresponding virtual channel id (except for setting the bit-fields/registers: CSI2_VC_CTRL.BTA_EN, CSI2_VC_TE.TE_SIZE, CSI2_VC_TE.TE_START, CSI2_VC_LONG_,,, CSI2_VC_SHORT_..., CSI2_VC_IRQ...). 1h (R/W) = Enabled. No change is allowed to the virtual channel registers expect resetting the VC_EN.

14.4.4.34 CSI2_VC_TE_0 to CSI2_VC_TE_3 Register (Offset = 104h + [i * 8h], where i = 0 to 3) [reset = 0h]

CSI2_VC_TE_0 to CSI2_VC_TE_3 is shown in Figure 14-192 and described in Table 14-188.

Return to Summary Table.

CONTROL REGISTER - Virtual channel This register controls the tearing effect logic. It defines the size of the transfer when TE occurs and enables the automatic TE mode.

Figure 14-192. CSI2_VC_TE_0 to CSI2_VC_TE_3 Register

31	30	29	28	27	26	25	24
TE_START	TE_EN	TE_LINE	TE_LINE_NB	RESERVED			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h			
23	22	21	20	19	18	17	16
TE_SIZE							
R/W-0h							
15	14	13	12	11	10	9	8
TE_SIZE							
R/W-0h							
7	6	5	4	3	2	1	0
TE_SIZE							
R/W-0h							

Table 14-188. CSI2_VC_TE_0 to CSI2_VC_TE_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TE_START	R/W	0h	Manual control of the start of the transfer. The user can use the TE interrupt in order to know that the TE trigger has been received prior to set the TE_START bit-field. It is not mandatory to use the TE interrupt. 0h (R/W) = Indicates the end of the transfer. The bit can be used by user to cancel the transfer if not already started. The FIFO shall be flushed by SW to ensure there is no data remaining in it. 1h (R/W) = Starts the transfer of the data. The size is defined in TE_SIZE. The bit-field is set until the transfer is completed. It is reset by HW when the transfer is completed.
30	TE_EN	R/W	0h	Tearing Effect Control 0h (R/W) = Disables the automatic transfer. The user shall use the interruption in order to know when TE PHY trigger is received or when the TE is detected on the input CMOS signals. The HW reset the bit-field when the transfer is completed (TE_SIZE=0). 1h (R/W) = Enables the automatic transfer of the data using the TE PHY trigger or one of the TE input signals as a synchronization event. The bit-field TE_LINE defines if the CMOS signal is used or if the PHY trigger is used.
29	TE_LINE	R/W	0h	
28	TE_LINE_NB	R/W	0h	Selection between TE0 and TE1 cmos signals. 0h (R/W) = TE0 cmos input line is selected 1h (R/W) = TE1 cmos input line is selected
27-24	RESERVED	R-0	0h	
23-0	TE_SIZE	R/W	0h	Defines the number of byte (payload data excluding the check -sum) to be sent. The write into the register CSI2_VC_LONG_PACKET_HEADER shall be performed by the user before sending data from the register CSI2_VC_LONG_PACKET_PAYLOAD. The register value is decremented for every byte sent of the CSI2 link. At the end of the transfer (TE_SIZE=0), the bit-field TE_EN is reset by HW. The DMA_request shall be asserted when the trigger is received in order to receive data in the TX FIFO. It shall not be used until all data (TE_SIZE) have been received in the FIFO.

14.4.4.35 CSI2_VC_LONG_PACKET_HEADER_0 to CSI2_VC_LONG_PACKET_HEADER_3 Register (Offset = 108h + [i * 8h], where i = 0 to 3) [reset = 0h]

CSI2_VC_LONG_PACKET_HEADER_0 to CSI2_VC_LONG_PACKET_HEADER_3 is shown in Figure 14-193 and described in Table 14-189.

Return to Summary Table.

LONG PACKET HEADER INFORMATION -Virtual channel
This register sets the 32-bit DATA_ID + Word count + ECC (the virtual channel id can be different than VC).
The ECC will be computed if ECC_TX_EN is set to 1.
DATA_ID is located at bit[7:0]
WC is located at bit[23:8]
ECC is located at bit[31:24]
(Least significant byte first and least significant bit first)

Figure 14-193. CSI2_VC_LONG_PACKET_HEADER_0 to CSI2_VC_LONG_PACKET_HEADER_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HEADER																															
W-0h																															

Table 14-189. CSI2_VC_LONG_PACKET_HEADER_0 to CSI2_VC_LONG_PACKET_HEADER_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HEADER	W	0h	Packet header information: DATA ID + DATA FIELD +ECC

14.4.4.36 CSI2_VC_LONG_PACKET_PAYLOAD_0 to CSI2_VC_LONG_PACKET_PAYLOAD_3 Register (Offset = 10Ch + [i * 8h], where i = 0 to 3) [reset = 0h]

CSI2_VC_LONG_PACKET_PAYLOAD_0 to CSI2_VC_LONG_PACKET_PAYLOAD_3 is shown in Figure 14-194 and described in Table 14-190.

Return to Summary Table.

LONG PACKET PAYLOAD INFORMATION -Virtual channel This register sets the payload information (excluding Check-sum). The HW shall capture the word count in the packet header (in CSI2_VC_LONG_PACKET_HEADER) in order to determine the last valid data. (the virtual channel id can be different than VC). Byte1 is bit[7:0] Byte2 is bit[15:8] Byte3 is bit[23:16] Byte4 is bit[31:24] Byten is sent before Byten+1 (Least significant byte first and least significant bit first)

Figure 14-194. CSI2_VC_LONG_PACKET_PAYLOAD_0 to CSI2_VC_LONG_PACKET_PAYLOAD_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAYLOAD																															
W-0h																															

Table 14-190. CSI2_VC_LONG_PACKET_PAYLOAD_0 to CSI2_VC_LONG_PACKET_PAYLOAD_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PAYLOAD	W	0h	Packet payload information (excluding check-sum)

14.4.4.37 CSI2_VC_SHORT_PACKET_HEADER_0 to CSI2_VC_SHORT_PACKET_HEADER_3 Register (Offset = 110h + [i * 8h], where i = 0 to 3) [reset = 0h]

CSI2_VC_SHORT_PACKET_HEADER_0 to CSI2_VC_SHORT_PACKET_HEADER_3 is shown in Figure 14-195 and described in Table 14-191.

Return to Summary Table.

SHORT PACKET HEADER INFORMATION -Virtual channel
This register sets the 24-bit DATA_ID + Short Packet Data Field + ECC (the virtual channel id can be different than VC).
The ECC will be computed if ECC_TX_EN is set to 1.
DATA_ID is located at bit[7:0]
Short Packet Data field is located at bit[23:8]
ECC is located at bit[31:24]
(Least significant byte first and least significant bit first)

Figure 14-195. CSI2_VC_SHORT_PACKET_HEADER_0 to CSI2_VC_SHORT_PACKET_HEADER_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HEADER																															
R/W-0h																															

Table 14-191. CSI2_VC_SHORT_PACKET_HEADER_0 to CSI2_VC_SHORT_PACKET_HEADER_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HEADER	R/W	0h	WRITES: Packet header information: DATA ID + DATA FIELD +ECC written into the TX FIFO READS: 32-bit values read from the RX FIFO

14.4.4.38 CSI2_VC_IRQSTATUS_0 to CSI2_VC_IRQSTATUS_3 Register (Offset = 118h + [i * 8h], where i = 0 to 3) [reset = 0h]

CSI2_VC_IRQSTATUS_0 to CSI2_VC_IRQSTATUS_3 is shown in Figure 14-196 and described in Table 14-192.

Return to Summary Table.

INTERRUPT STATUS REGISTER - Virtual channel This register regroups all the events related to the virtual channel.

Figure 14-196. CSI2_VC_IRQSTATUS_0 to CSI2_VC_IRQSTATUS_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							PP_BUSY_CHANGE_IRQ
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
FIFO_TX_UDF_IRQ	ECC_NO_CORRECTION_IRQ	BTA_IRQ	FIFO_RX_OVF_IRQ	FIFO_TX_OVF_IRQ	PACKET_SENT_IRQ	ECC_CORRECTION_IRQ	CS_IRQ
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-192. CSI2_VC_IRQSTATUS_0 to CSI2_VC_IRQSTATUS_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	PP_BUSY_CHANGE_IRQ	R/W	0h	Video Port ping-pong buffer busy status. PP_BUSY has changed from 1 to 0. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
7	FIFO_TX_UDF_IRQ	R/W	0h	FIFO underflow status. The FIFO used on the slave port for buffering the data received on the OCP slave port for the virtual channel has underflowed which means that the data for the current packet have not been received in time since the transfer of the packet are already started (transfer started since the packet size is bigger than space allocated in the FIFO). 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
6	ECC_NO_CORRECTION_IRQ	R/W	0h	ECC error status (short and long packets). No correction of the header because of more than 1-bit error. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
5	BTA_IRQ	R/W	0h	Virtual channel - BTA status. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.

Table 14-192. CSI2_VC_IRQSTATUS_0 to CSI2_VC_IRQSTATUS_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	FIFO_RX_OVF_IRQ	R/W	0h	FIFO overflow error status. The FIFO used on the slave port for buffering the data received on the CSI2 link for the virtual channel has overflowed. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
3	FIFO_TX_OVF_IRQ	R/W	0h	FIFO overflow error status. The FIFO used on the slave port for buffering the data received on the OCP slave port for the virtual channel has overflowed. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
2	PACKET_SENT_IRQ	R/W	0h	Indicates that a packet has been sent. It is used when BTA manual mode is used. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
1	ECC_CORRECTION_IRQ	R/W	0h	Virtual channel - ECC has been used to do the correction of the only 1-bit error status (short and long packet). 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.
0	CS_IRQ	R/W	0h	Virtual channel - Check-Sum mismatch status. 0h (R/W) = READS: Event is false. WRITES: Status bit unchanged. 1h (R/W) = READS: Event is true (pending). WRITES: Status bit is reset.

14.4.4.39 CSI2_VC_IRQENABLE_0 to CSI2_VC_IRQENABLE_3 Register (Offset = 11Ch + [i * 8h], where i = 0 to 3) [reset = 0h]

CSI2_VC_IRQENABLE_0 to CSI2_VC_IRQENABLE_3 is shown in Figure 14-197 and described in Table 14-193.

Return to Summary Table.

INTERRUPT ENABLE REGISTER - Virtual channel This register regroups all the events related to virtual channel.

Figure 14-197. CSI2_VC_IRQENABLE_0 to CSI2_VC_IRQENABLE_3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							PP_BUSY_CHANGE_IRQ_EN
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
FIFO_TX_UDF_IRQ_EN	ECC_NO_CORRECTION_IRQ_EN	BTA_IRQ_EN	FIFO_RX_OVF_IRQ_EN	FIFO_TX_OVF_IRQ_EN	PACKET_SENT_IRQ_EN	ECC_CORRECTION_IRQ_EN	CS_IRQ_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 14-193. CSI2_VC_IRQENABLE_0 to CSI2_VC_IRQENABLE_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R-0	0h	
8	PP_BUSY_CHANGE_IRQ_EN	R/W	0h	Video Port ping-pong buffer busy status. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
7	FIFO_TX_UDF_IRQ_EN	R/W	0h	FIFO underflow enable. The FIFO used on the slave port for buffering the data received on the OCP slave port for the virtual channel has underflowed which means that the data for the current packet have not been received in time since the transfer of the packet are already started (transfer started since the packet size is bigger than space allocated in the FIFO). 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
6	ECC_NO_CORRECTION_IRQ_EN	R/W	0h	ECC error (short and long packets). No correction of the header because of more than 1-bit error. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
5	BTA_IRQ_EN	R/W	0h	Virtual channel -Bus turn around reception 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
4	FIFO_RX_OVF_IRQ_EN	R/W	0h	FIFO overflow enable. The FIFO used on the slave port for buffering the data received on the CSI2 link for the virtual channel has overflowed. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs

Table 14-193. CSI2_VC_IRQENABLE_0 to CSI2_VC_IRQENABLE_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	FIFO_TX_OVF_IRQ_EN	R/W	0h	FIFO overflow enable. The FIFO used on the slave port for buffering the data received on the OCP slave port for the virtual channel has overflowed. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
2	PACKET_SENT_IRQ_EN	R/W	0h	Indicates that a packet has been sent. It is used when BTA manual mode is used. 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
1	ECC_CORRECTION_IRQ_EN	R/W	0h	Virtual channel - ECC has been used to correct the only 1-bit error (short and long packet). 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs
0	CS_IRQ_EN	R/W	0h	Virtual channel - Check-Sum of the payload mismatch detection 0h (R/W) = Event is masked 1h (R/W) = Event generates an interrupt when it occurs

14.4.5 CSI2_PHY Registers

Table 14-194 lists the memory-mapped registers for the CSI2_PHY. All register offset addresses not listed in Table 14-194 should be considered as reserved locations and the register contents should not be modified.

Table 14-194. CSI2_PHY Registers

Offset	Acronym	Register Name	Section
0h	REGISTER0	First Register	Section 14.4.5.1
4h	REGISTER1		Section 14.4.5.2
8h	REGISTER2		Section 14.4.5.3
Ch	REGISTER3		Section 14.4.5.4
10h	REGISTER4		Section 14.4.5.5
14h	REGISTER5		Section 14.4.5.6
18h	REGISTER6		Section 14.4.5.7
1Ch	REGISTER7		Section 14.4.5.8
20h	REGISTER8		Section 14.4.5.9
24h	REGISTER9		Section 14.4.5.10
28h	REGISTER10		Section 14.4.5.11
2Ch	REGISTER11		Section 14.4.5.12
30h	REGISTER12		Section 14.4.5.13
34h	REGISTER13		Section 14.4.5.14
38h	REGISTER14		Section 14.4.5.15
3Ch	REGISTER15		Section 14.4.5.16

Complex bit access types are encoded to fit into small table cells. Table 14-195 shows the codes that are used for access types in this section.

Table 14-195. CSI2_PHY Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	-0 R	Returns 0s Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

14.4.5.1 REGISTER0 Register (Offset = 0h) [reset = 1E481D3Ah]

REGISTER0 is shown in Figure 14-198 and described in Table 14-196.

Return to Summary Table.

First Register

Figure 14-198. REGISTER0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_THSPREPARE								REG_THSPRPR_THSZERO							
R/W-1Eh								R/W-48h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_THSTRAIL								REG_THSEXIT							
R/W-1Dh								R/W-3Ah							

Table 14-196. REGISTER0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	REG_THSPREPARE	R/W	1Eh	REG_THSPREPARE timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4. D-PHY spec:40ns+4UI-85ns+6UI Auctual value seen on line : = REG_THSPREPARE timer + analog delay & slew on signals = REG_THSPREPARE +(-26.5ns - - +4 ns) PROGRAMMED VALUE = ceil(70 ns / DDR Clock Period) + 2 Default value is programmed for 400 MHz
23-16	REG_THSPRPR_THSZERO	R/W	48h	REG_THSPREPARE_THSZERO timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4. D-PHY spec: > 145ns + 10 UI Actual value seen on line N= REG_THSPREPARE_THSZERO - REG_THSPREPARE M=REG_THSPREPARE = [ceil [(N+3)/4] * 4 + ceil [M/4] * 4 + 3] * DDR_Clock_Period+[~-29ns - 0ns] PROGRAMMED VALUE = ceil (175 ns/ DDR Clock Period) + 2 Default value is programmed for 400 MHz
15-8	REG_THSTRAIL	R/W	1Dh	REG_THSTRAIL timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4. D-PHY spec : > 60ns + 4UI Actual value seen on line : N = REG_THSTRAIL = [ceil [(N+3)/4] * 4 - 2.75] * DDR_Clock_period +(~ 0ns - 5ns) PROGRAMMED VALUE = ceil (60 ns / DDR Clock Period) + 5 Default value is programmed for 400 MHz
7-0	REG_THSEXIT	R/W	3Ah	REG_THSEXIT timing parameter in multiples of DDR clock frequency. DDR clock = CLKIN4DDR/4. D-PHy spec : > 100 ns Actual value seen on line : N = REG_THSEXIT = THSEXIT timer + analog delay & slew on LP signals = [ceil[N/4] * 4] * DDR_CLOCK_PERIOD - (~3 ns - 45 ns) PROGRAMMED VALUE = ceil (145 ns/ DDR Clock Period) Default value is programmed for 400 MHz

14.4.5.2 REGISTER1 Register (Offset = 4h) [reset = 420A1A6Ah]

REGISTER1 is shown in Figure 14-199 and described in Table 14-197.

Return to Summary Table.

Figure 14-199. REGISTER1 Register

31	30	29	28	27	26	25	24
REG_TTAGO			REG_TTASURE		REG_TTAGET		
R/W-2h			R/W-0h		R/W-2h		
23	22	21	20	19	18	17	16
EMPTY			REG_TLPXBY2				
R-0-0h			R/W-Ah				
15	14	13	12	11	10	9	8
REG_TCLKTRAIL							
R/W-1Ah							
7	6	5	4	3	2	1	0
REG_TCLKZERO							
R/W-6Ah							

Table 14-197. REGISTER1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	REG_TTAGO	R/W	2h	TTA-GO timing in terms of number of TXCLKESC clocks 000 == 2 cycles 001 == 3 cycles 010 == 4 cycles 011 == 5 cycles 100 == 6 cycles 101 == 7 cycles 110 == 8 cycles 111 == 9 cycles Default value: 4 cycles
28-27	REG_TTASURE	R/W	0h	TTA-SURE timing in terms of number of TXCLKESC clocks 00 == 2 cycles 01 == Reserved 10 == 3 cycles 11 == 4 cycles Default value: 2 cycles
26-24	REG_TTAGET	R/W	2h	TTA-GET timing in terms of number of TXCLKESC clocks 000 == 3 cycles 001 == 4 cycles 010 == 5 cycles 011 == 6 cycles 100 == 7 cycles 101 == 8 cycles 110 == 9 cycles 111 == 10 cycles Default value: 5 cycles
23-21	EMPTY	R-0	0h	Reserved
20-16	REG_TLPXBY2	R/W	Ah	(TLPX)/2 timing parameter in multiples of DDR clock frequency. DDR clock = CLKIN4DDR/4. PROGRAMMED VALUE = ceil (25 ns / DDR Clock Period) Default value is programmed for 400 MHz Note : TLPX is used to define the length of LP-01 state in HS start of Transmision sequences on clock and data lanes. For all other purposes TLPX is defined by the period of TXLPESC
15-8	REG_TCLKTRAIL	R/W	1Ah	REG_TCLKTRAIL timing parameter in multiples of DDR clock frequency. DDR clock = CLKIN4DDR/4. D-PHY spec : 60 ns PROGRAMMED VALUE = ceil (60 ns / DDR Clock Period) Actual value seen on line: N = REG_TCLKTRAIL = [ceil[(N+3)/4] *4 - 1.5] * DDR_Clock_Period + (~0 ns - 5 ns) PROGRAMMED VALUE = ceil (60 ns / DDR Clock Period) + 2 Default value is programmed for 400 MHz
7-0	REG_TCLKZERO	R/W	6Ah	REG_TCLKZERO timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4. D-PHY spec :[REG_TCLKPREPARE + REG_TCLKZERO] > 300 ns Derived spec for REG_TCLKERO (Min REG_TCLKPREPARE = 38 ns) REG_TCLKZERO > 262 ns Actual value seen on line : N = REG_TCLKZERO M = REG_TCLKPREPARE = [ceil [(N+3)/4] *4 + ceil (M/4) * 4 - M + 2] * DDR_Clock_Period + [~0ns - 5ns] PROGRA MMED VALUE = ceil (265 ns / DDR Clock Period) Default value is programmed for 400 MHz

14.4.5.3 REGISTER2 Register (Offset = 8h) [reset = B800001Ah]

REGISTER2 is shown in Figure 14-200 and described in Table 14-198.

Return to Summary Table.

Figure 14-200. REGISTER2 Register

31	30	29	28	27	26	25	24
HSSYNCPATTERN							
R/W-B8h							
23	22	21	20	19	18	17	16
DATARATE	OVRRDLPTXGZ	REGLPTXGZ					OVRRDULPMTX
R/W-0h	R/W-0h	R/W-0h					R/W-0h
15	14	13	12	11	10	9	8
REGULPMTX					EMPTY		
R/W-0h					R/W-0h		
7	6	5	4	3	2	1	0
REG_TCLKPREPARE							
R/W-1Ah							

Table 14-198. REGISTER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	HSSYNCPATTERN	R/W	B8h	Default : 184 (10111000). MSB (last received bit of sync pattern), LSB (first received bit of sync pattern).
23	DATARATE	R/W	0h	This bit is set high if data rate < 400Mbps => DDR clock < 200MHz. 1 : Will put High-Speed Transmitters in power-saving mode. Default value : 0
22	OVRRDLPTXGZ	R/W	0h	1: Override with register 0: Default
21-17	REGLPTXGZ	R/W	0h	1: Tri-stated 0: Not Tri-stated
16	OVRRDULPMTX	R/W	0h	1: Override with register 0: Default
15-11	REGULPMTX	R/W	0h	1: Transmit ULPM 0: Don't transmit ULPM
10-8	EMPTY	R/W	0h	Reserved
7-0	REG_TCLKPREPARE	R/W	1Ah	D-PHY spec : 38ns - 95 ns Actual value seen online : = REG_TCLKPREPARE timer + analog delay & slew on LP signals = REG_TCLKPREPARE * DDR Clock Period + [~25 ns -+ 5 ns) PROGRAMMED VALUE = ceil (65 ns / DDR Clock Period) Default value is programmed for 400 MHz

14.4.5.4 REGISTER3 Register (Offset = Ch) [reset = 625D21A0h]

REGISTER3 is shown in Figure 14-201 and described in Table 14-199.

Return to Summary Table.

Figure 14-201. REGISTER3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_TXTRIGGERESC3								REG_TXTRIGGERESC2							
R/W-62h								R/W-5Dh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_TXTRIGGERESC1								REG_TXTRIGGERESC0							
R/W-21h								R/W-A0h							

Table 14-199. REGISTER3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	REG_TXTRIGGERESC3	R/W	62h	Transmitted pattern when REG_TXTRIGGERESC3 is asserted (first bit transmitted to last bit transmitted). Default: 01100010
23-16	REG_TXTRIGGERESC2	R/W	5Dh	Default: 01011101
15-8	REG_TXTRIGGERESC1	R/W	21h	Default: 00100001
7-0	REG_TXTRIGGERESC0	R/W	A0h	Default: 10100000

14.4.5.5 REGISTER4 Register (Offset = 10h) [reset = 625D21A0h]

REGISTER4 is shown in Figure 14-202 and described in Table 14-200.

Return to Summary Table.

Figure 14-202. REGISTER4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_RXTRIGGERESC3								REG_RXTRIGGERESC2							
R/W-62h								R/W-5Dh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_RXTRIGGERESC1								REG_RXTRIGGERESC0							
R/W-21h								R/W-A0h							

Table 14-200. REGISTER4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	REG_RXTRIGGERESC3	R/W	62h	Received pattern for which REG_RXTRIGGERESC3 is asserted (first bit transmitted to last bit transmitted). Default: 01100010
23-16	REG_RXTRIGGERESC2	R/W	5Dh	Default: 01011101
15-8	REG_RXTRIGGERESC1	R/W	21h	Default: 00100001
7-0	REG_RXTRIGGERESC0	R/W	A0h	Default: 10100000

14.4.5.6 REGISTER5 Register (Offset = 14h) [reset = 0h]

REGISTER5 is shown in Figure 14-203 and described in Table 14-201.

Return to Summary Table.

Figure 14-203. REGISTER5 Register

31	30	29	28	27	26	25	24
RESETDONETXBYTECLK	RESETDONESCPCLK	RESETDONEPWRCLK	RESETDONETXCLKESC4	RESETDONETXCLKESC3	RESETDONETXCLKESC2	RESETDONETXCLKESC1	RESETDONETXCLKESC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
EMPTY							
R-0h							
15	14	13	12	11	10	9	8
EMPTY							
R-0h							
7	6	5	4	3	2	1	0
EMPTY							
R-0h							

Table 14-201. REGISTER5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESETDONETXBYTECLK	R	0h	RESETDONETXBYTECLK
30	RESETDONESCPCLK	R	0h	RESETDONESCPCLK
29	RESETDONEPWRCLK	R	0h	RESETDONEPWRCLK
28	RESETDONETXCLKESC4	R	0h	RESETDONETXCLKESC4
27	RESETDONETXCLKESC3	R	0h	RESETDONETXCLKESC3
26	RESETDONETXCLKESC2	R	0h	RESETDONETXCLKESC2
25	RESETDONETXCLKESC1	R	0h	RESETDONETXCLKESC1
24	RESETDONETXCLKESC0	R	0h	RESETDONETXCLKESC0
23-0	EMPTY	R	0h	Reserved

14.4.5.7 REGISTER6 Register (Offset = 18h) [reset = 00010000h]

REGISTER6 is shown in Figure 14-204 and described in Table 14-202.

Return to Summary Table.

Figure 14-204. REGISTER6 Register

31	30	29	28	27	26	25	24
OVRRDHSTXEN	REGHSTXEN					OVRRDHSTXTERMEN	REGHSTXTERMEN
R/W-0h	R/W-0h					R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
REGHSTXTERMEN				OVRRDCLKLANEADDR	REGCLKLANEADDR		
R/W-0h				R/W-0h	R/W-1h		
15	14	13	12	11	10	9	8
OVRRDDEEMPDISABLE	REGDEEMPDISABLE					OVRRDLPTXEN	REGLPTXEN
R/W-0h	R/W-0h					R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
REGLPTXEN				EMPTY		BYPASSCOMPFILT	BYPASSCOMP
R/W-0h				R-0-0h		R/W-0h	R/W-0h

Table 14-202. REGISTER6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	OVRRDHSTXEN	R/W	0h	1: Override with register 0: Default
30-26	REGHSTXEN	R/W	0h	1: Enable 0: Disable
25	OVRRDHSTXTERMEN	R/W	0h	1: Override with register 0: Default
24-20	REGHSTXTERMEN	R/W	0h	1: Enable 0: Disable
19	OVRRDCLKLANEADDR	R/W	0h	1: Override with register 0: Default
18-16	REGCLKLANEADDR	R/W	1h	001: Lane0 -> clock lane 010: Lane1-> clock lane 011: Lane2 -> clock lane 101: Lane3 -> clock lane Other states are reserved. Default: 001
15	OVRRDDEEMPDISABLE	R/W	0h	1: Override with register bit 0: Default
14-10	REGDEEMPDISABLE	R/W	0h	1:Disable De-emphasis 0:Enable De-emphasis
9	OVRRDLPTXEN	R/W	0h	1: Override with register 0: Default
8-4	REGLPTXEN	R/W	0h	1: Enable 0: Disable
3-2	EMPTY	R-0	0h	Reserved
1	BYPASSCOMPFILT	R/W	0h	1 : Bypass LP-RX and LP-CD with fast buffers 0 : Do not bypass
0	BYPASSCOMP	R/W	0h	1 : Bypass LP-RX and LP-CD comparator with fast buffers 0 : Do not bypass

14.4.5.8 REGISTER7 Register (Offset = 1Ch) [reset = 0h]

REGISTER7 is shown in Figure 14-205 and described in Table 14-203.

Return to Summary Table.

Figure 14-205. REGISTER7 Register

31	30	29	28	27	26	25	24
OVRRDLPRXEN	REGLPRXEN					OVRRDLPCDEN	REGLPCDEN
R/W-0h	R/W-0h					R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
REGLPCDEN				OVRRDULPRXEN	REGULPRXEN		
R/W-0h				R/W-0h	R/W-0h		
15	14	13	12	11	10	9	8
REGULPRXEN		OVRRDLDOVDDTRACKING	REGLDOVDDTRACKING	OVRRDHSLDOOBSERVE	REGHSLDOOBSERVE	OVRRDBIASGENTRIMMODE	REGBIASGENTRIMMODE
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
OVRRDHSTXCOREEN	REGHSTXCOREEN					EMPTY	
R/W-0h	R/W-0h					R-0-0h	

Table 14-203. REGISTER7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	OVRRDLPRXEN	R/W	0h	1: Override with register 0: Default
30-26	REGLPRXEN	R/W	0h	1:Enable 0:Disable
25	OVRRDLPCDEN	R/W	0h	1: Override with register 0: Default
24-20	REGLPCDEN	R/W	0h	1:Enable 0:Disable
19	OVRRDULPRXEN	R/W	0h	1: Override with register 0: Default
18-14	REGULPRXEN	R/W	0h	1:Enable 0:Disable
13	OVRRDLDOVDDTRACKING	R/W	0h	1: Override with register 0: Default
12	REGLDOVDDTRACKING	R/W	0h	1: Enable LDO tracking VDD 0: Disable(default)
11	OVRRDHSLDOOBSERVE	R/W	0h	1: Override with register 0: Default
10	REGHSLDOOBSERVE	R/W	0h	1: Enable HS LDO observe 0: Disable
9	OVRRDBIASGENTRIMMODE	R/W	0h	1: Override with register 0: Default
8	REGBIASGENTRIMMODE	R/W	0h	1: Enable 0: Disable
7	OVRRDHSTXCOREEN	R/W	0h	1: Override with register 0: Default
6-2	REGHSTXCOREEN	R/W	0h	1: Enable 0: Disable(default)
1-0	EMPTY	R-0	0h	Reserved

14.4.5.9 REGISTER8 Register (Offset = 20h) [reset = 410h]

REGISTER8 is shown in Figure 14-206 and described in Table 14-204.

Return to Summary Table.

Figure 14-206. REGISTER8 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				OVRRDHSTXTERMRES	REGHSTXTERMRES		
R-0h				R/W-0h	R/W-10h		
7	6	5	4	3	2	1	0
REGHSTXTERMRES		OVRRDEFUSEBIASGEN	BIASGENCODE				
R/W-10h		R/W-0h	R/W-10h				

Table 14-204. REGISTER8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11	OVRRDHSTXTERMRES	R/W	0h	1: Override EFUSE bits 0: Use EFUSE bits
10-6	REGHSTXTERMRES	R/W	10h	11111: Max resistance 10000: Nominal resistance 00000: Min resistance
5	OVRRDEFUSEBIASGEN	R/W	0h	1: Override EFUSE bits 0: Use EFUSE bits
4-0	BIASGENCODE	R/W	10h	11111: Min current 10000: Nominal current 00000: Max current

14.4.5.10 REGISTER9 Register (Offset = 24h) [reset = 00020001h]

REGISTER9 is shown in Figure 14-207 and described in Table 14-205.

Return to Summary Table.

Figure 14-207. REGISTER9 Register

31	30	29	28	27	26	25	24
OVRRDPOLARITY	REGPOLARITY3TO0					EMPTY	OVRRDBIASGENEN
R/W-0h	R/W-0h					R-0h	R/W-0h
23	22	21	20	19	18	17	16
REGBIASGENEN	OVRRDBANDGAPEN	REGBANDGAPEN	ENBIASGENCURROUT	ENLDOVOLTAGECONTROL	REGLDOVOLTAGE		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-4h		
15	14	13	12	11	10	9	8
REGLDOVOLTAGE	ENBIASGENCONTROL	REGBIASGENTESTMODES		ENVREGCONTROL	REGVREGLOAD		REGVREGBIASCURR
R/W-4h	R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
REGVREGBIASCURR	ENLPTXIMPBYPASS	REGLPTXIMPBYPASS	OVRRDCLKIN4DDRSIGNALS	REGBYPASSEN	REGCLKINEN	REGCLKIN4DDRGODDBAR	REGBYPASSACKZ
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 14-205. REGISTER9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	OVRRDPOLARITY	R/W	0h	1: Override with register bit 0: Default
30-26	REGPOLARITY3TO0	R/W	0h	1: DX=DN, DY=DP 0: DX=DP, DY=DN
25	EMPTY	R	0h	Reserved
24	OVRRDBIASGENEN	R/W	0h	1: Override with Register bit 0 : Default
23	REGBIASGENEN	R/W	0h	1: Enable 0: Disable
22	OVRRDBANDGAPEN	R/W	0h	1: Override with Register bit 0: Default
21	REGBANDGAPEN	R/W	0h	1: Enable 0: Disable
20	ENBIASGENCURROUT	R/W	0h	1: Enable biasgen current (10uA) brought out. 0: Biasgen current is not brought out.(Note this testmode will be used only in testchips. Not expected to be used in SOC)
19	ENLDOVOLTAGECONTROL	R/W	0h	1: Control LDO output voltage through register bits. 0: Set LDO voltage to default
18-15	REGLDOVOLTAGE	R/W	4h	1111: Maximum voltage 0000: Minimum voltage Default: 0100
14	ENBIASGENCONTROL	R/W	0h	1: Control Bias current through register bits. 0: Set bias current to default.
13-12	REGBIASGENTESTMODES	R/W	0h	Default: 00
11	ENVREGCONTROL	R/W	0h	1: Enable control. 0: Use default current
10-9	REGVREGLOAD	R/W	0h	Default: 00
8-7	REGVREGBIASCURR	R/W	0h	Default: 00
6	ENLPTXIMPBYPASS	R/W	0h	1: LP-TX bypass controlled by register bit. 0: Default value
5	REGLPTXIMPBYPASS	R/W	0h	1: output impedance bypassed permanently 0: output impedance not bypassed at all
4	OVRRDCLKIN4DDRSIGNALS	R/W	0h	1: Override signals with register settings 0: Default
3	REGBYPASSEN	R/W	0h	BYPASSEN

Table 14-205. REGISTER9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	REGCLKINEN	R/W	0h	CLKINEN
1	REGCLKIN4DDRGODDBAR	R/W	0h	CLKIN4DDRGOODBAR
0	REGBYPASSACKZ	R/W	1h	BYPASSACKZ

14.4.5.11 REGISTER10 Register (Offset = 28h) [reset = 0h]

REGISTER10 is shown in Figure 14-208 and described in Table 14-206.

Return to Summary Table.

Figure 14-208. REGISTER10 Register

31	30	29	28	27	26	25	24
LDOWAKEUPTIME		LPPOWERUPTIME		ENLPTXSCPDAT	REGLPTXSCPDAT4TO0DXDY		
R/W-0h		R/W-0h		R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
REGLPTXSCPDAT4TO0DXDY							ENHSTXSCPDAT
R/W-0h							R/W-0h
15	14	13	12	11	10	9	8
REGHSTXSCPDAT4TO0					OVRRDLDOEN	REGLDOEN	
R/W-0h					R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
REGLDOEN				EMPTY			
R/W-0h				R-0h			

Table 14-206. REGISTER10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	LDOWAKEUPTIME	R/W	0h	Set LPTX wakeup time counter in PWRCLK cycles or TXCLKESC cycles. In PWRCLK cycles (when PWRCMD is changed). 00: 250 (Default) 01: 313 10: 375 11: 500
29-28	LPPOWERUPTIME	R/W	0h	LDO wakeup time counter in number of PWRCLK cycles (approximate) 00: 12500 (Default) 01: 6250 10: 31250 11: 62500
27	ENLPTXSCPDAT	R/W	0h	1: LPTX Data Taken from register 0: Default
26-17	REGLPTXSCPDAT4TO0DXDY	R/W	0h	26-17: Data for DX4,DY4,DX3,DY3, DX2,DY2, DX1,DY1, DX0,DY0 respectively
16	ENHSTXSCPDAT	R/W	0h	1: HS Data Taken from register 0: Default
15-11	REGHSTXSCPDAT4TO0	R/W	0h	15-11: Data for D4, D3, D2, D1 and D0 respectively
10	OVRRDLDOEN	R/W	0h	1: Override with Register bit 0: Default
9-4	REGLDOEN	R/W	0h	1: Enable 0: Disable
3-0	EMPTY	R	0h	Reserved

14.4.5.12 REGISTER11 Register (Offset = 2Ch) [reset = 7D20A59Ah]

REGISTER11 is shown in Figure 14-209 and described in Table 14-207.

Return to Summary Table.

Figure 14-209. REGISTER11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOOPBACKDATABYTE3								LOOPBACKDATABYTE2							
R/W-7Dh								R/W-20h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOOPBACKDATABYTE1								LOOPBACKDATABYTE0							
R/W-A5h								R/W-9Ah							

Table 14-207. REGISTER11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LOOPBACKDATABYTE3	R/W	7Dh	Fourth byte transmitted in loop-back mode
23-16	LOOPBACKDATABYTE2	R/W	20h	Third byte transmitted in loop-back mode
15-8	LOOPBACKDATABYTE1	R/W	A5h	Second byte transmitted in loop-back mode
7-0	LOOPBACKDATABYTE0	R/W	9Ah	First byte transmitted in loop-back mode

14.4.5.13 REGISTER12 Register (Offset = 30h) [reset = 04320000h]

REGISTER12 is shown in Figure 14-210 and described in Table 14-208.

Return to Summary Table.

Figure 14-210. REGISTER12 Register

31	30	29	28	27	26	25	24
TCLKPRE							
R/W-4h							
23	22	21	20	19	18	17	16
TCLKPOST							
R/W-32h							
15	14	13	12	11	10	9	8
OVRRDLANEENABLE	REGLANEENABLE					EMPTY	
R/W-0h	R/W-0h					R-0-0h	
7	6	5	4	3	2	1	0
BGTRIMBITS							
R/W-0h							

Table 14-208. REGISTER12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TCLKPRE	R/W	4h	TCLK-PRE used in loop-back mode - in number of DDR clocksDefault: 4 DDRCLK cycles
23-16	TCLKPOST	R/W	32h	TCLK-POST used in loop-back mode - in number of DDR clocksDefault: 50 DDRCLK cycles
15	OVRRDLANEENABLE	R/W	0h	Default: 0
14-10	REGLANEENABLE	R/W	0h	Default: 00000
9-8	EMPTY	R-0	0h	Reserved
7-0	BGTRIMBITS	R/W	0h	Maximum: 100000 Minimum: 011111 Default: 000000

14.4.5.14 REGISTER13 Register (Offset = 34h) [reset = 0h]

REGISTER13 is shown in Figure 14-211 and described in Table 14-209.

Return to Summary Table.

Figure 14-211. REGISTER13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANALOGTESTMODES																								EMPTY							
R/W-0h																								R-0h							

Table 14-209. REGISTER13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	ANALOGTESTMODES	R/W	0h	Default:0
7-0	EMPTY	R	0h	Reserved

14.4.5.15 REGISTER14 Register (Offset = 38h) [reset = 3E080000h]

REGISTER14 is shown in Figure 14-212 and described in Table 14-210.

Return to Summary Table.

Figure 14-212. REGISTER14 Register

31	30	29	28	27	26	25	24
OVRRDBGCONTROL	REGBGCONTROL						
R/W-0h	R/W-7Ch						
23	22	21	20	19	18	17	16
REGBGCONTROL	OVRRDHSDELAYCALIBEN	REGHSDELAYCALIBEN	OVRRDHSDELAYCALIBCLRZ	REGHSDELAYCALIBCLRZ	OVRRDHSTXDELAYMASTERCTRL	REGHSTXDELAYMASTERCTRL	
R/W-7Ch	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
REGHSTXDELAYMASTERCTRL		LPCDSAMPLEOUTEN	LPRXSAMPLEOUTEN	OVRRDEFUSEBG	OVRRDHSTXDELAYSLAVECTRL	REGHSTXDELAYSLAVECTRL	
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
REGHSTXDELAYSLAVECTRL		EMPTY					
R/W-0h		R-0h					

Table 14-210. REGISTER14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	OVRRDBGCONTROL	R/W	0h	1: Override bandgap control bits with register value 0: Default
30-23	REGBGCONTROL	R/W	7Ch	Default: 01111100
22	OVRRDHSDELAYCALIBEN	R/W	0h	1: Override with register bit 0: Default
21	REGHSDELAYCALIBEN	R/W	0h	1: Enable 0: Disable
20	OVRRDHSDELAYCALIBCLRZ	R/W	0h	1: Override with register bit 0: Default
19	REGHSDELAYCALIBCLRZ	R/W	1h	1: Clrz disabled(default) 0. Clrz enable
18	OVRRDHSTXDELAYMASTERCTRL	R/W	0h	1: Override with register bit 0: Default
17-14	REGHSTXDELAYMASTERCTRL	R/W	0h	Default: 0000
13	LPCDSAMPLEOUTEN	R/W	0h	1: Enable sampled data to be brought out on WPO pins. 0: Normal mode
12	LPRXSAMPLEOUTEN	R/W	0h	1: Enable sampled data to be brought out on WPO pins. 0: Normal mode
11	OVRRDEFUSEBG	R/W	0h	1: Override EFUSE bits 0: Use EFUSE bits
10	OVRRDHSTXDELAYSLAVECTRL	R/W	0h	1: Override with register bit 0: Default
9-6	REGHSTXDELAYSLAVECTRL	R/W	0h	Default: 0000
5-0	EMPTY	R	0h	Reserved

14.4.5.16 REGISTER15 Register (Offset = 3Ch) [reset = 01400003h]

REGISTER15 is shown in Figure 14-213 and described in Table 14-211.

Return to Summary Table.

Figure 14-213. REGISTER15 Register

31	30	29	28	27	26	25	24
OVRRDLDOHIZEN	REGLDOHIZEN						REG_THSTXEN
R/W-0h	R/W-0h						R/W-Ah
23	22	21	20	19	18	17	16
REG_THSTXEN			EMPTY				
R/W-Ah			R-0h				
15	14	13	12	11	10	9	8
EMPTY							NOTRANSITIONDISABLE
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NOTRANSITIONDISABLE				OVRRDNOTRANSITIONCTRL	REGNOTRANSITIONCTRL		
R/W-0h				R/W-0h	R/W-3h		

Table 14-211. REGISTER15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	OVRRDLDOHIZEN	R/W	0h	1: Override with register bit 0: Default
30-25	REGLDOHIZEN	R/W	0h	1: LDO in Hiz Mode 0: LDO not in Hiz mode
24-21	REG_THSTXEN	R/W	Ah	REG_THSTXEN is timing parameter for HSTXEN deassertion staggering. staggered time = REG_THSTXEN * (4 * Reg_Tlpxby2) Default : 1010 1000 ns 0000 : Reserved
20-9	EMPTY	R	0h	Reserved
8-4	NOTRANSITIONDISABLE	R/W	0h	1: NoTransition disabled 0: NoTransition controlled by FSM Default: 00000
3	OVRRDNOTRANSITIONCTRL	R/W	0h	1: Override with register bits 0: Default
2-0	REGNOTRANSITIONCTRL	R/W	3h	Default: 011

Hardware Accelerator

Topic **Page**

15.1 Introduction **2114**
15.2 Register Map **2114**

15.1 Introduction

The radar hardware accelerator is a hardware IP that enables offloading the burden of certain frequently used computations in FMCW radar signal processing from the main processor. FMCW radar signal processing involves the use of FFT and Log-Magnitude computations to obtain a radar image across the range, velocity, and angle dimensions. Some of the frequently used functions in FMCW radar signal processing can be done within the radar hardware accelerator, while still retaining the flexibility of implementing other proprietary algorithms in the main processor. See the *Hardware Accelerator User's Guide* (SWRU526 for part 1 and SWRU527 part 2) for a functional description and features of this module.

The following describes which Hardware Accelerator (HWA) version is present in each silicon revision of each device:

	HWA V1.0	HWA v1.05	HWA V1.1
Device	IWR1443 ES1.0, ES2.0, ES3.0	IWR6843 ES1.0	IWR6843 ES2.0

15.2 Register Map

For details on the hardware accelerator register, see the *Radar Hardware Accelerator User's Guide*.

Real Time Interrupt (RTI) and RTI With Digital Watchdog Timer (WDT)

This section describes the functionality of the real-time interrupt (RTI) module. The RTI is designed as an operating system timer to support a real time operating system (RTOS). The watchdog timer is also implemented, using another instance of the RTI module.

NOTE: This chapter describes a superset implementation of the RTI module that includes features and functionality related to DMA, and Timebase control. These features are dependent on the device-specific feature content. Consult your device-specific data sheet to determine the applicability of these features to the device being used.

Topic **Page**

16.1 Overview **2116**
16.2 Module Operation **2117**
16.3 RTI Control Registers **2124**

16.1 Overview

The real-time interrupt (RTI) module provides timer functionality for operating systems and for benchmarking code. The RTI module can incorporate several counters that define the timebases needed for scheduling in the operating system.

The timers also allow you to benchmark certain areas of code by reading the values of the counters at the beginning and the end of the desired code range and calculating the difference between the values.

16.1.1 Features

The RTI module has the following features:

- Two independent 64 bit counter blocks
- Four configurable compares for generating operating system ticks or DMA requests. Each event can be driven by either counter block 0 or counter block 1.
- Fast enabling/disabling of events
- Two time stamp (capture) functions for system or peripheral interrupts, one for each counter block
- Digital windowed watchdog

16.1.2 Industry Standard Compliance Statement

This module is specifically designed to fulfill the requirements for OSEK (**O**ffene **S**ysteme und deren **S**chnittstellen für die **E**lektronik im **K**raftfahrzeug, or Open Systems and the Corresponding Interfaces for Automotive Electronics) as well as OSEK/time-compliant operating systems, but is not limited to it.

16.2 Module Operation

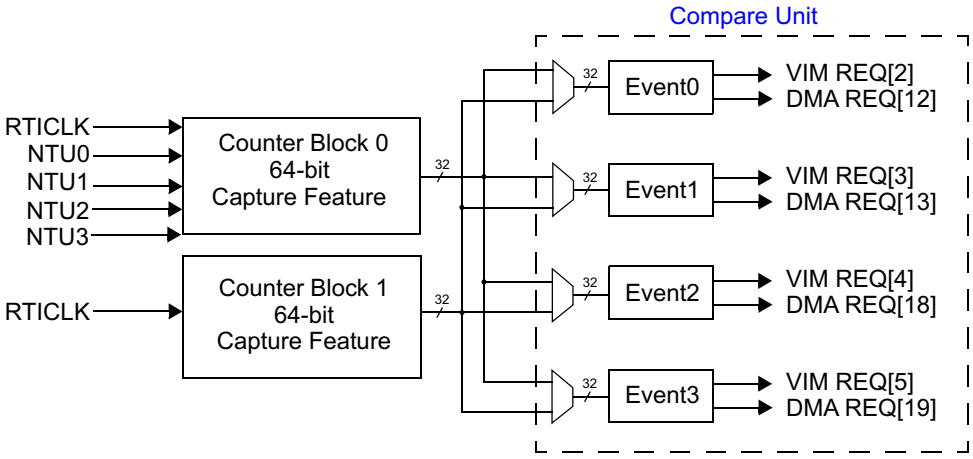
Figure 16-1 illustrates the high level block diagram of the RTI module.

The RTI module has two independent counter blocks for generating different timebases: counter block 0 and counter block 1.

A compare unit compares the counters with programmable values and generates four independent interrupt or DMA requests on compare matches. Each of the compare registers can be programmed to be compared to either counter block 0 or counter block 1.

The following sections describe the individual functions in more detail.

Figure 16-1. RTI Block Diagram



16.2.1 Counter Operation

Each counter block consists of the following (see Figure 16-2):

- One 32-bit prescale counter (RTIUC0 or RTIUC1)
- One 32-bit free running counter (RTIFRC0 or RTIFRC1)

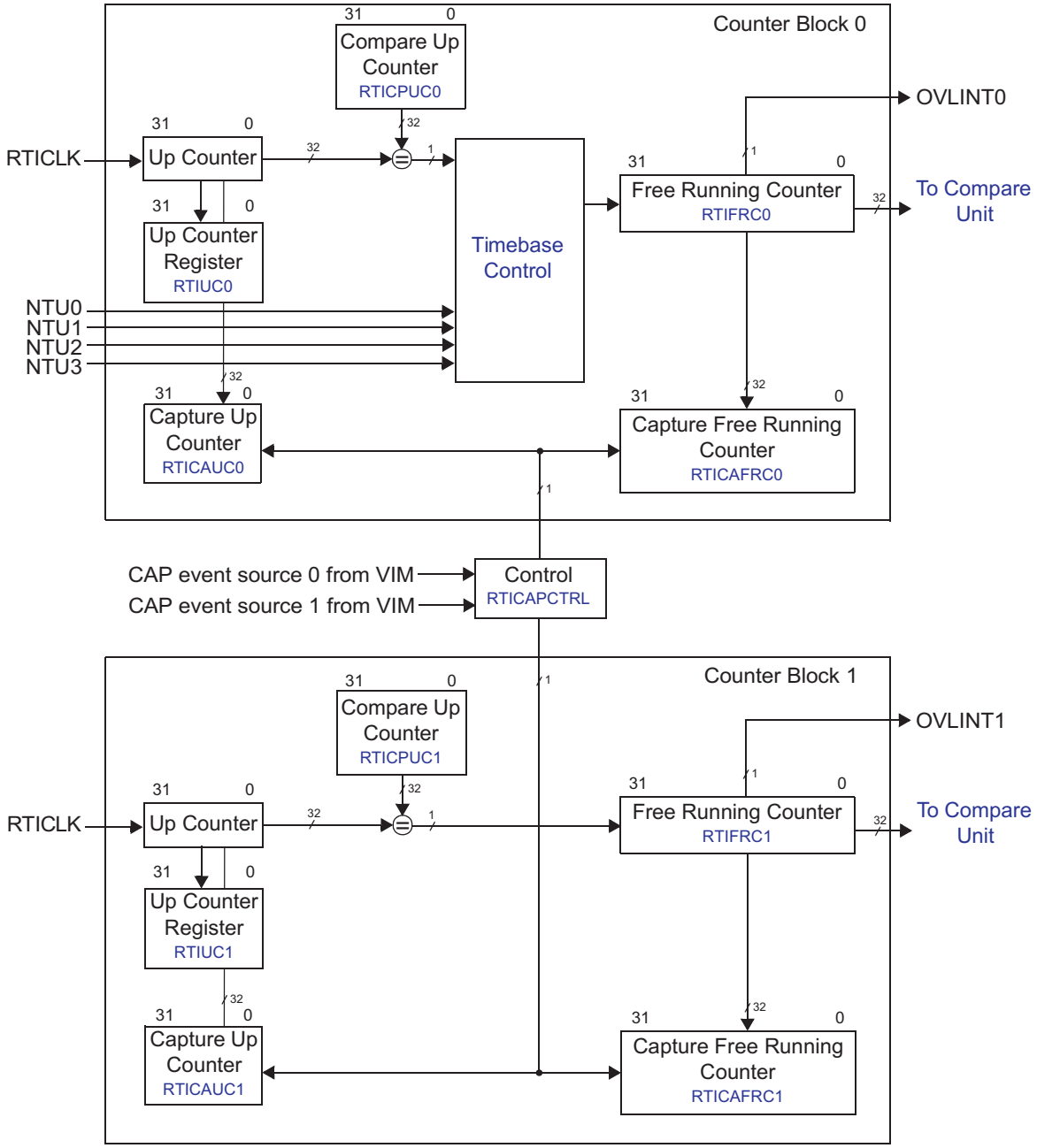
The RTIUC0/1 is driven by the RTICLK and counts up until the compare value in the compare up counter register (RTICPUC0 or RTICPUC1) is reached. When the compare matches, RTIFRC0/1 is incremented and RTIUC0/1 is reset to 0. If RTIFRC0/1 overflows, an interrupt is generated to the vectored interrupt manager (VIM). The overflow interrupt is not intended to generate the timebase for the operating system. See Section 16.2.2 for the timebase generation. The up counter together with the compare up counter value prescale the RTI clock. The resulting formula for the frequency of the free running counter (RTIFRC0/1) is:

$$f_{RTIFRCx} = \begin{cases} \dfrac{f_{RTICLK}}{RTICPUCx + 1} & \text{when RTICPUCx} \neq 0 \\ \dfrac{f_{RTICLK}}{2^{32}+1} & \text{when RTICPUCx} = 0 \end{cases} \quad (3)$$

NOTE: Setting RTICPUCx equal to zero is not recommended. Doing so will hold the Up Counter at zero for two RTICLK cycles after it overflows from 0xFFFFFFFF to zero.

The counter values can be determined by reading the respective counter registers or by generating a hardware event which captures the counter value into the respective capture register. Both functions are described in the following sections.

Figure 16-2. Counter Block Diagram



16.2.1.1 Counter and Capture Read Consistency

Portions of the device internal databus are 32-bits wide. If the application wants to read the 64-bit counters or the 64-bit capture values, a certain order of 32-bit read operations needs to be followed. This is to prevent one counter incrementing in between the two separate read operations to both counters.

Reading the Counters

The free running counter (RTIFRCx) must be read first. This priority will ensure that in the cycle when the CPU reads RTIFRCx, the up counter value is stored in its counter register (RTIUCx). The second read has to access the up counter register (RTIUCx), which then holds the value which corresponds to the number of RTICLK cycles that have elapsed at the time reading the free running counter register (RTIFRCx).

NOTE: The up counters are implemented as shadow registers. Reading RTIUCx without having read RTIFRCx first will return always the same value. RTIUCx will only be updated when RTIFRCx is read.

Reading the Capture Values

The free running counter capture register (RTICAFRCx) must be read first. This priority will ensure that in the cycle when the CPU reads RTICAFRCx, the up counter value is stored in its counter register (RTICAUCx). The second read has to access the up counter register (RTICAUCx), which then holds the value captured at the time when reading the capture free running counter register (RTICAFRCx).

NOTE: The capture up counter registers are implemented as shadow registers. Reading RTICAUCx without having read RTICAFRCx first will return always the same value. RTICAUCx will only be updated when RTICAFRCx is read.

16.2.1.2 Capture Feature

Both counter blocks also provide a capture feature on external events. Two capture sources can trigger the capture event. The source triggering the block is configurable (RTICAPCTRL). The sources originate from the Vectored Interrupt Manager (VIM) and allow the generation of capture events when a peripheral modules has generated an interrupt. Any of the peripheral interrupts can be selected as the capture event in the VIM.

When an event is detected, RTIUCx and RTIFRCx are stored in the capture up counter (RTICAUCx) and capture free running counter (RTICAFRCx) registers. The read order of the captured values must be the same as the read order of the actual counters (see Section 16.2.1.1).

16.2.2 Interrupt/DMA Requests

There are four compare registers (RTICOMPy) to generate interrupt requests to the VIM or DMA requests to the DMA controller. The interrupts can be used to generate different timebases for the operating system. Each of the compare registers can be configured to be compared to either RTIFRC0 or RTIFRC1. When the counter value matches the compare value, an interrupt is generated. To allow periodic interrupts, a certain value can be added to the compare value in RTICOMPy automatically. This value is stored in the update compare register (RTIUDCPy) and will be added after a compare is matched. The period of the generated interrupt/DMA request can be calculated with:

$$t_{COMPx} = t_{RTICLK} \times (RTICPUCy + 1) \times RTIUDCPy$$

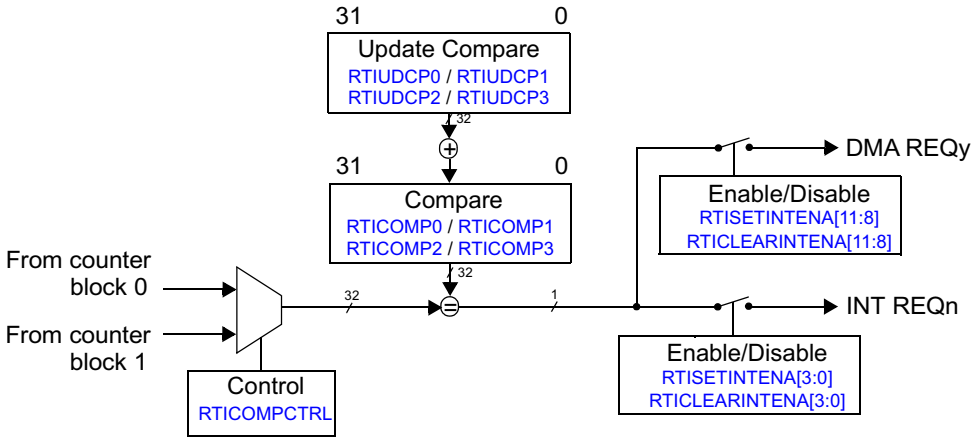
if RTICPUCy = 0,

$$t_{COMPx} = t_{RTICLK} \times (2^{32}+1) \times RTIUDCPy$$

if RTIUDCPy = 0,

$$t_{COMPx} = t_{RTICLK} \times (RTICPUCy + 1) \times 2^{32} \quad (4)$$

Figure 16-3. Compare Unit Block Diagram (shows only 1 of 4 blocks for simplification)



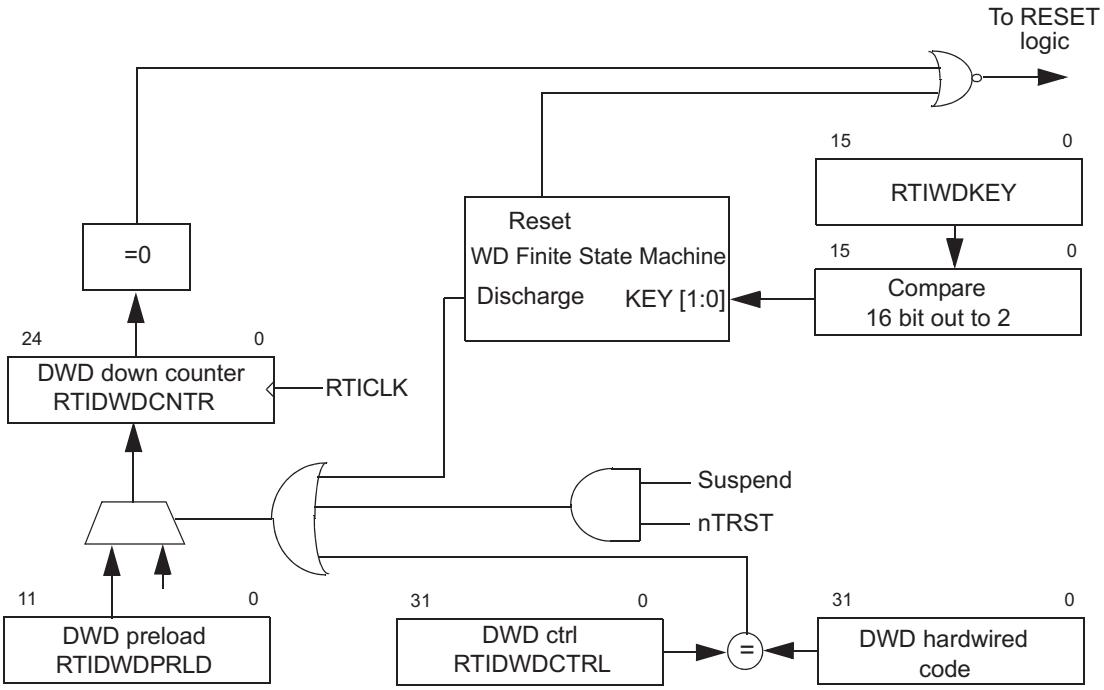
Another interrupt that can be generated is the overflow interrupt (OVLINTx) in case the RTIFRCx counter overflows.

The interrupts/DMA requests can be enabled in the RTISETINTENA register and disabled in the RTICLEARINTENA register. The RTIINTFLAG register shows the pending interrupts.

16.2.3 Digital Watchdog (DWD)

The digital watchdog (DWD) is an optional safety diagnostic which can detect a runaway CPU and generate either a reset or NMI (non-maskable interrupt) response. It generates resets or NMIs after a programmable period, or if no correct key sequence was written to the RTIWDKEY register. Figure 16-4 illustrates the DWD.

Figure 16-4. Digital Watchdog



16.2.3.1 Digital Watchdog (DWD)

The DWD is disabled by default. If it should be used, it must be enabled by writing a 32-bit value to the RTIDWDCTRL register.

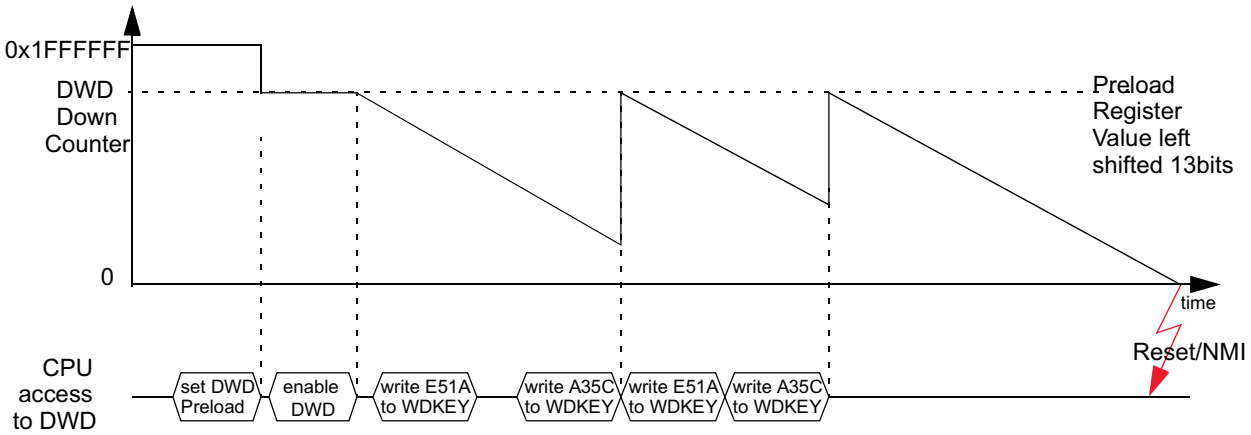
NOTE: Once the DWD is enabled, it cannot be disabled except by system reset or power on reset.

If the correct key sequence is written to the RTIWDKEY register (0xE51A followed by 0xA35C), the 25-bit DWD down counter is reloaded with the left justified 12-bit preload value stored in RTIDWDPRLD. If an incorrect value is written, a watchdog reset or NMI will occur immediately. A reset or NMI will also be generated when the DWD down counter is decremented to 0.

While the device is in suspend mode (halting debug mode), the DWD down counter keeps the value it had when entering suspend mode.

The DWD down counter will be decremented with the RTICLK frequency.

Figure 16-5. DWD Operation



The expiration time of the DWD down counter can be determined with the following equation:

$texp = (DWDPRLD + 1) \times 2^{13}/RTICLK$

where

DWDPRLD = 0...4095

NOTE: Care should be taken to ensure that the CPU write to the watchdog register is made allowing time for the write to propagate to the RTI.

16.2.3.2 Digital Windowed Watchdog (DWWD)

In addition to the time-out boundary configurable via the digital watchdog discussed in Section 16.2.3.1, for enhanced safety metrics it is desirable to check for a watchdog "pet" within a time window rather than using a single time threshold. This is enabled by the digital windowed watchdog (DWWD) feature.

- Functional Behavior

The DWWD opens a configurable time window in which the watchdog must be serviced. Any attempt to service the watchdog outside this time window, or a failure to service the watchdog in this time window, will cause the watchdog to generate either a reset or a NMI to the CPU. This is controlled by configuring the RTIWWDRXNCTRL register. As with the DWD, the DWWD is disabled after power on reset. When the DWWD is configured to generate a non-maskable interrupt on a window violation, the watchdog counter continues to count down. The NMI handler needs to clear the watchdog violation status flag(s) and then

service the watchdog by writing the correct sequence in the watchdog key register. This service will cause the watchdog counter to get reloaded from the preload value and start counting down. If the NMI handler does not service the watchdog in time, it could count down all the way to zero and wrap around. If the NMI Handler does not service the watchdog in time, the NMI gets generated continuously, each time the counter counts to '0'.

The DWWD uses the Digital Watchdog (DWD) preload register (RTIDWDPRLD) setting to define the end-time of the window. The start-time of the window is defined by a window size configuration register(RTIWWDSIZECTRL).

The default window size is set to 100%, which corresponds to the DWD functionality of a time-out-only watchdog. The window size can be selected (through register RTIWWDSIZECTRL) from among 100%, 50%, 25%, 12.5%, 6.25% and 3.125% as shown in Figure 16-6. The window with the respective size will be opened before the end of the DWD expiration. The user has to serve the watchdog in the window. Otherwise, a reset or NMI will generate. Figure 16-7 shows an DWWD operation example (25% window).

- Configuration of DWWD

The DWWD preload value (same as DWD preload) can only be configured when the DWWD counter is disabled. The window size and watchdog reaction to a violation can be configured even after the watchdog has been enabled. Any changes to the window size and watchdog reaction configurations will only take effect after the next servicing of the DWWD. This feature can be utilized to dynamically set windows of different sizes based on task execution time, adding a program sequence element to the diagnostic which can improve fault coverage.

Figure 16-6. Digital Windowed Watchdog Timing Example

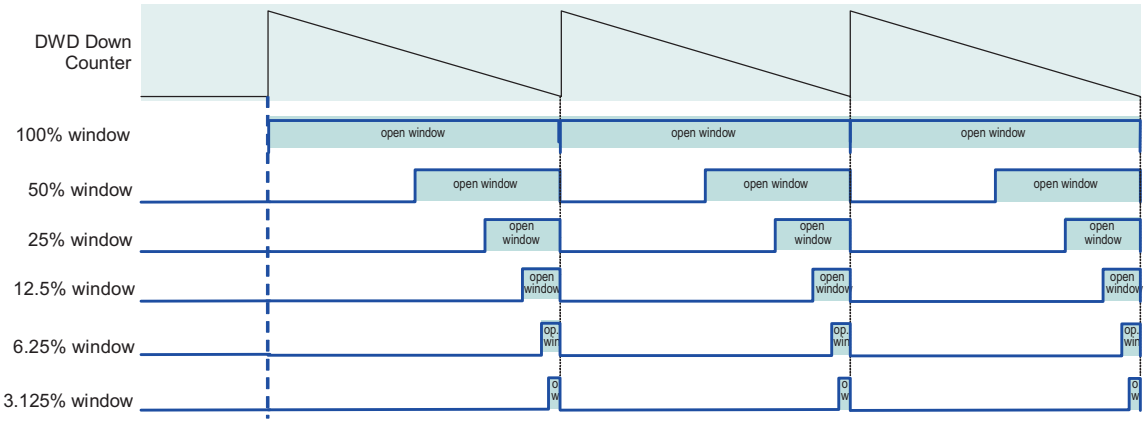
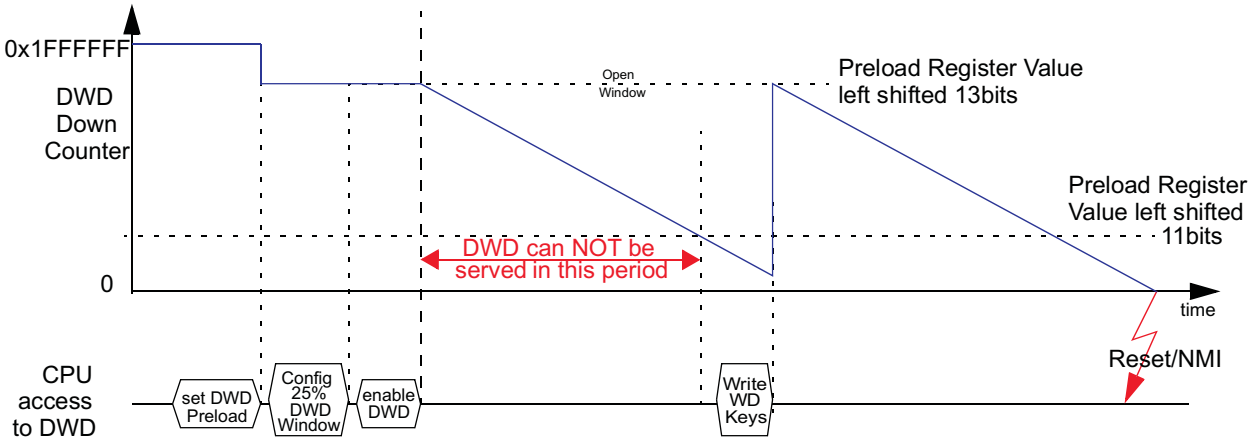


Figure 16-7. Digital Windowed Watchdog Operation Example (25% Window)



16.2.4 Halting Debug Mode Behaviour

Once the system enters halting debug mode, the behavior of the RTI depends on the COS (continue on suspend) bit. If the bit is cleared and halting debug mode is active, all counters will stop operation. If the bit is set to one, all counters will be clocked normally and the RTI will work like in normal mode.

16.3 RTI Control Registers

Table 16-1 provides a summary of the registers. The registers support 8-bit, 16-bit, and 32-bit writes. The offset is relative to the associated peripheral select. See the following sections for detailed descriptions of the registers. The subsystem contains two RTI modules, RTI1 and RTI2. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

Table 16-1. RTI Registers

Offset	Acronym	Register Description	Section
00h	RTIGCTRL	RTI Global Control Register	Section 16.3.1
04h	RTITBCTRL	RTI Timebase Control Register	Section 16.3.2
08h	RTICAPCTRL	RTI Capture Control Register	Section 16.3.3
0Ch	RTICOMPCTRL	RTI Compare Control Register	Section 16.3.4
10h	RTIFRC0	RTI Free Running Counter 0 Register	Section 16.3.5
14h	RTIUC0	RTI Up Counter 0 Register	Section 16.3.6
18h	RTICPUC0	RTI Compare Up Counter 0 Register	Section 16.3.7
20h	RTICAFRC0	RTI Capture Free Running Counter 0 Register	Section 16.3.8
24h	RTICAUC0	RTI Capture Up Counter 0 Register	Section 16.3.9
30h	RTIFRC1	RTI Free Running Counter 1 Register	Section 16.3.10
34h	RTIUC1	RTI Up Counter 1 Register	Section 16.3.11
38h	RTICPUC1	RTI Compare Up Counter 1 Register	Section 16.3.12
40h	RTICAFRC1	RTI Capture Free Running Counter 1 Register	Section 16.3.13
44h	RTICAUC1	RTI Capture Up Counter 1 Register	Section 16.3.14
50h	RTICOMP0	RTI Compare 0 Register	Section 16.3.15
54h	RTIUDCP0	RTI Update Compare 0 Register	Section 16.3.16
58h	RTICOMP1	RTI Compare 1 Register	Section 16.3.17
5Ch	RTIUDCP1	RTI Update Compare 1 Register	Section 16.3.18
60h	RTICOMP2	RTI Compare 2 Register	Section 16.3.19
64h	RTIUDCP2	RTI Update Compare 2 Register	Section 16.3.20
68h	RTICOMP3	RTI Compare 3 Register	Section 16.3.21
6Ch	RTIUDCP3	RTI Update Compare 3 Register	Section 16.3.22
70h	RTITBLCOMP	RTI Timebase Low Compare Register	Section 16.3.23
74h	RTITBHCOMP	RTI Timebase High Compare Register	Section 16.3.24
80h	RTISETINTENA	RTI Set Interrupt Enable Register	Section 16.3.25
84h	RTICLEARINTENA	RTI Clear Interrupt Enable Register	Section 16.3.26
88h	RTIINTFLAG	RTI Interrupt Flag Register	Section 16.3.27
90h	RTIDWDCTRL	Digital Watchdog Control Register	Section 16.3.28
94h	RTIDWDPRLD	Digital Watchdog Preload Register	Section 16.3.29
98h	RTIWDSTATUS	Watchdog Status Register	Section 16.3.30
9Ch	RTIWDKEY	RTI Watchdog Key Register	Section 16.3.31
A0h	RTIDWDCNTR	RTI Digital Watchdog Down Counter Register	Section 16.3.32
A4h	RTIWWDRXNCTRL	Digital Windowed Watchdog Reaction Control Register	Section 16.3.33
A8h	RTIWWDSIZECTRL	Digital Windowed Watchdog Window Size Control Register	Section 16.3.34
ACh	RTIINTCLRENABLE	RTI Compare Interrupt Clear Enable Register	Section 16.3.35
B0h	RTICOMP0CLR	RTI Compare 0 Clear Register	Section 16.3.36
B4h	RTICOMP1CLR	RTI Compare 1 Clear Register	Section 16.3.37
B8h	RTICOMP2CLR	RTI Compare 2 Clear Register	Section 16.3.38
BCh	RTICOMP3CLR	RTI Compare 3 Clear Register	Section 16.3.39

NOTE: Writes to Reserved registers may clear the pending RTI interrupt.

16.3.1 RTI Global Control Register (RTIGCTRL)

The global control register starts/stops the counters and selects the signal compared with the timebase control circuit. This register is shown in Figure 16-8 and described in Table 16-2.

Figure 16-8. RTI Global Control Register (RTIGCTRL) [offset = 00]

31	20	19	16
Reserved		NTUSEL	
R-0		R/WP-0	

15	14	2	1	0
COS	Reserved		CNT1EN	CNT0EN
R/WP-0	R-0		R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-2. RTI Global Control Register (RTIGCTRL) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reads return 0. Writes have no effect.
19-16	NTUSEL		Select NTU signal. These bits determine which NTU input signal is used as external timebase
		0h	NTU0
		5h	NTU1
		Ah	NTU2
		Fh	NTU3
		All other values	Tied to 0
15	COS		Continue on suspend. This bit determines if both counters are stopped when the device goes into halting debug mode or if they continue counting.
		0	Counters are stopped while in halting debug mode.
		1	Counters are running while in halting debug mode.
14-2	Reserved	0	Reads return 0. Writes have no effect.
1	CNT1EN		Counter 1 enable. This bit starts and stops counter block 1 (RTIUC1 and RTIFRC1).
		0	Counter block 1 is stopped.
		1	Counter block 1 is running.
0	CNT0EN		Counter 0 enable. This bit starts and stops counter block 0 (RTIUC0 and RTIFRC0).
		0	Counter block 0 is stopped.
		1	Counter block 0 is running.

NOTE: If the application uses the timebase circuit for synchronization between the communications controller and the operating system and the device enters halting debug mode, the synchronization may be lost depending on the COS setting in the RTI module and the halting debug mode behavior of the communications controller.

16.3.2 RTI Timebase Control Register (RTITBCTRL)

The timebase control register selects if the free running counter 0 is incremented by RTICLK or NTU. This register is shown in Figure 16-9 and described in Table 16-3.

Figure 16-9. RTI Timebase Control Register (RTITBCTRL) [offset = 04h]

31	8
Reserved	
R-0	

7	2	1	0
Reserved		INC	TBEXT
R-0		R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-3. RTI Timebase Control Register (RTITBCTRL) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reads return 0. Writes have no effect.
1	INC		Increment free running counter 0. This bit determines whether the free running counter 0 (RTIFRC0) is automatically incremented if a failing clock on the NTU signal is detected.
		0	RTIFRC0 will not be incremented on a failing external clock.
		1	RTIFRC0 will be incremented on a failing external clock.
0	TBEXT		Timebase external. This bit selects whether the free running counter 0 (RTIFRC0) is clocked by the internal up counter 0 (RTIUC0) or from the external signal NTU. Setting the TBEXT bit from 0 to 1 will not increment RTIFRC0, since RTIUC0 is reset.
			When the timebase supervisor circuit detects a missing clock edge, then the TBEXT bit is reset.
			Only the software can select whether the external signal should be used.
		0	RTIUC0 clocks RTIFRC0.
		1	NTU clocks RTIFRC0.

16.3.3 RTI Capture Control Register (RTICAPCTRL)

The capture control register controls the capture source for the counters. This register is shown in Figure 16-10 and described in Table 16-4.

Figure 16-10. RTI Capture Control Register (RTICAPCTRL) [offset = 08h]

31	8
Reserved	
R-0	

7	2	1	0
Reserved		CAPCNTR1	CAPCNTR0
R-0		R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-4. RTI Capture Control Register (RTICAPCTRL) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reads return 0. Writes have no effect.
1	CAPCNTR1		Capture counter 1. This bit determines which external interrupt source triggers a capture event of RTIUC1 and RTIFRC1.
		0	Capture of RTIUC1/ RTIFRC1 is triggered by capture event source 0.
		1	Capture of RTIUC1/ RTIFRC1 is triggered by capture event source 1.
0	CAPCNTR0		Capture counter 0. This bit determines which external interrupt source triggers a capture event of RTIUC0 and RTIFRC0.
		0	Capture of RTIUC0/ RTIFRC0 is triggered by capture event source 0.
		1	Capture of RTIUC0/ RTIFRC0 is triggered by capture event source 1.

16.3.4 RTI Compare Control Register (RTICOMPCTRL)

The compare control register controls the source for the compare registers. This register is shown in Figure 16-11 and described in Table 16-5.

Figure 16-11. RTI Compare Control Register (RTICOMPCTRL) [offset = 0Ch]

31	16
Reserved	
R-0	

15	13	12	11	9	8
Reserved		COMPSEL3	Reserved		COMPSEL2
R-0		R/WP-0	R-0		R/WP-0

7	5	4	3	1	0
Reserved		COMPSEL1	Reserved		COMPSEL0
R-0		R/WP-0	R-0		R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-5. RTI Compare Control Register (RTICOMPCTRL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reads return 0. Writes have no effect.
12	COMPSEL3		Compare select 3. This bit determines the counter with which the compare value held in compare register 3 (RTICOMP3) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
11-9	Reserved	0	Reads return 0. Writes have no effect.
8	COMPSEL2		Compare select 2. This bit determines the counter with which the compare value held in compare register 2 (RTICOMP2) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
7-5	Reserved	0	Reads return 0. Writes have no effect.
4	COMPSEL1		Compare select 1. This bit determines the counter with which the compare value held in compare register 1 (RTICOMP1) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
3-1	Reserved	0	Reads return 0. Writes have no effect.
0	COMPSEL0		Compare select 0. This bit determines the counter with which the compare value held in compare register 0 (RTICOMP0) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.

16.3.5 RTI Free Running Counter 0 Register (RTIFRC0)

The free running counter 0 register holds the current value of free running counter 0. This register is shown in Figure 16-12 and described in Table 16-6.

Figure 16-12. RTI Free Running Counter 0 Register (RTIFRC0) [offset = 10h]

31	16
FRC0	
R/WP-0	

15	0
FRC0	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-6. RTI Free Running Counter 0 Register (RTIFRC0) Field Descriptions

Bit	Field	Value	Description
31-0	FRC0	0-FFFF FFFFh	Free running counter 0. This registers holds the current value of the free running counter 0. A read of this counter returns the current value of the counter. The counter can be preset by writing (in privileged mode only) to this register. The counter increments then from this written value upwards. **Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC0 and RTIFRC0.**

16.3.6 RTI Up Counter 0 Register (RTIUC0)

The up counter 0 register holds the current value of prescale counter. This register is shown in Figure 16-13 and described in Table 16-7.

Figure 16-13. RTI Up Counter 0 Register (RTIUC0) [offset = 14h]

31	16
UC0	
R/WP-0	

15	0
UC0	
R/WP-0	

LLEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-7. RTI Up Counter 0 Register (RTIUC0) Field Descriptions

Bit	Field	Value	Description
31-0	UC0	0-FFFF FFFFh	Up counter 0. This register holds the current value of the up counter 0 and prescales the RTI clock. It will be only updated by a previous read of free running counter 0 (RTIFRC0). This method of updating effectively gives a 64-bit read of both counters, without having the problem of a counter being updated between two consecutive reads on up counter 0 (RTIUC0) and free running counter 0 (RTIFRC0). A read of this counter returns the value of the counter at the time RTIFRC0 was read. A write to this counter presets it with a value. The counter then increments from this written value upwards. Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC0 and RTIFRC0. **Note: If the preset value is bigger than the compare value stored in register RTICPUC0, then it can take a long time until a compare matches, since RTIUC0 has to count up until it overflows.**

16.3.7 RTI Compare Up Counter 0 Register (RTICPUC0)

The compare up counter 0 register holds the value to be compared with prescale counter 0 (RTIUC0). This register is shown in Figure 16-14 and described in Table 16-8.

Figure 16-14. RTI Compare Up Counter 0 Register (RTICPUC0) [offset = 18h]

31	16
CPUC0	
R/W-0	

15	0
CPUC0	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16-8. RTI Compare Up Counter 0 Register (RTICPUC0) Field Descriptions

Bit	Field	Value	Description

31-0	CPUC0	0-FFFF FFFFh	Compare up counter 0. This register holds the value that is compared with the up counter 0. When the compare shows a match, the free running counter 0 (RTIFRC0) is incremented. RTIUC0 is set to 0 when the counter value matches the RTICPUC0 value. The value set in this register prescales the RTI clock.
			If CPUC0 = 0, then
			f_{FRC0} = RTICLK/(2^{32}+1) (Setting CPUC0 equal to 0 is not recommended. Doing so will hold the Up Counter at 0 for 2 RTICLK cycles after it overflows from FFFF FFFFh to 0.)
			If CPUC0 ≠ 0, then
			f_{FRC0} = RTICLK/(RTICPUC0+1)
			A read of this register returns the current compare value.
			A write to this register:
			• If TBEXT = 0, the compare value is updated.
			• If TBEXT = 1, the compare value is unchanged.

16.3.8 RTI Capture Free Running Counter 0 Register (RTICAFRC0)

The capture free running counter 0 register holds the free running counter 0 on external events. This register is shown in Figure 16-15 and described in Table 16-9.

Figure 16-15. RTI Capture Free Running Counter 0 Register (RTICAFRC0) [offset = 20h]

31	16
CAFRC0	
R-0	

15	0
CAFRC0	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16-9. RTI Capture Free Running Counter 0 Register (RTICAFRC0) Field Descriptions

Bit	Field	Value	Description
31-0	CAFRC0	0-FFFF FFFFh	Capture free running counter 0. This register captures the current value of the free running counter 0 (RTIFRC0) when an event occurs, controlled by the external capture control block. A read of this register returns the value of RTIFRC0 on a capture event.

16.3.9 RTI Capture Up Counter 0 Register (RTICAUC0)

The capture up counter 0 register holds the current value of prescale counter 0 on external events. This register is shown in Figure 16-16 and described in Table 16-10.

Figure 16-16. RTI Capture Up Counter 0 Register (RTICAUC0) [offset = 24h]

31	16
CAUC0	
R-0	

15	0
CAUC0	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16-10. RTI Capture Up Counter 0 Register (RTICAUC0) Field Descriptions

Bit	Field	Value	Description
31-0	CAUC0	0-FFFF FFFFh	Capture up counter 0. This register captures the current value of the up counter 0 (RTIUC0) when an event occurs, controlled by the external capture control block. **Note: The read sequence must be the same as with RTIUC0 and RTIFRC0. Therefore, the RTICAFRC0 register must be read before the RTICAUC0 register is read. This sequence ensures that the value of the RTICAUC0 register is the corresponding value to the RTICAFRC0 register, even if another capture event happens in between the two reads.** A read of this register returns the value of RTIUC0 on a capture event.

16.3.10 RTI Free Running Counter 1 Register (RTIFRC1)

The free running counter 1 register holds the current value of the free running counter 1. This register is shown in Figure 16-17 and described in Table 16-11.

Figure 16-17. RTI Free Running Counter 1 Register (RTIFRC1) [offset = 30h]

31	16
FRC1	
R/WP-0	

15	0
FRC1	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-11. RTI Free Running Counter 1 Register (RTIFRC1) Field Descriptions

Bit	Field	Value	Description
31-0	FRC1	0-FFFF FFFFh	Free running counter 1. This register holds the current value of the free running counter 1 and will be updated continuously. A read of this register returns the current value of the counter. A write to this register presets the counter. The counter increments then from this written value upwards. **Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC1 and RTIFRC1.**

16.3.11 RTI Up Counter 1 Register (RTIUC1)

The up counter 1 register holds the current value of the prescale counter 1. This register is shown in Figure 16-18 and described in Table 16-12.

Figure 16-18. RTI Up Counter 1 Register (RTIUC1) [offset = 34h]

31	16
UC1	
R/WP-0	

15	0
UC1	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-12. RTI Up Counter 1 Register (RTIUC1) Field Descriptions

Bit	Field	Value	Description
31-0	UC1	0-FFFF FFFFh	Up counter 1. This register holds the current value of the up counter 1 and prescales the RTI clock. It will be only updated by a previous read of free running counter 1 (RTIFRC1). This method of updating effectively gives a 64-bit read of both counters, without having the problem of a counter being updated between two consecutive reads on RTIUC1 and RTIFRC1. A read of this register will return the value of the counter when the RTIFRC1 was read. A write to this register presets the counter. The counter then increments from this written value upwards. **Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC1 and RTIFRC1.** **Note: If the preset value is bigger than the compare value stored in register RTICPUC1, then it can take a long time until a compare matches, since RTIUC1 has to count up until it overflows.**

16.3.12 RTI Compare Up Counter 1 Register (RTICPUC1)

The compare up counter 1 register holds the value compared with prescale counter 1. This register is shown in Figure 16-19 and described in Table 16-13.

Figure 16-19. RTI Compare Up Counter 1 Register (RTICPUC1) [offset = 38h]

31	16
CPUC1	
R/WP-0	

15	0
CPUC1	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-13. RTI Compare Up Counter 1 Register (RTICPUC1) Field Descriptions

Bit	Field	Value	Description

31-0	CPUC1	0-FFFF FFFFh	Compare up counter 1. This register holds the compare value, which is compared with the up counter 1. When the compare matches, the free running counter 1 (RTIFRC1) is incremented. The up counter is cleared to 0 when the counter value matches the CPUC1 value. The value set in this prescales the RTI clock according to the following formula:
			If CPUC1 = 0, then
			f_{FRC1} = RTICLK/(2^{32}+1) (Setting CPUC1 equal to 0 is not recommended. Doing so will hold the Up Counter at 0 for 2 RTICLK cycles after it overflows from FFFF FFFFh to 0.)
			If CPUC1 ≠ 0, then
			f_{FRC1} = RTICLK/(RTICPUC1+1)
			A read of this register returns the current compare value.
			A write to this register updates the compare value.

16.3.13 RTI Capture Free Running Counter 1 Register (RTICAFRC1)

The capture free running counter 1 register holds the current value of free running counter 1 on external events. This register is shown in Figure 16-20 and described in Table 16-14.

Figure 16-20. RTI Capture Free Running Counter 1 Register (RTICAFRC1) [offset = 40h]

31	16
CAFRC1	
R-0	

15	0
CAFRC1	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16-14. RTI Capture Free Running Counter 1 Register (RTICAFRC1) Field Descriptions

Bit	Field	Value	Description
31-0	CAFRC1	0-FFFF FFFFh	Capture free running counter 1. This register captures the current value of the free running counter 1 (RTIFRC1) when an event occurs, controlled by the external capture control block. A read of this register returns the value of RTIFRC1 on a capture event.

16.3.14 RTI Capture Up Counter 1 Register (RTICAUC1)

The capture up counter 1 register holds the current value of prescale counter 1 on external events. This register is shown in Figure 16-21 and described in Table 16-15.

Figure 16-21. RTI Capture Up Counter 1 Register (RTICAUC1) [offset = 44h]

31	16
CAUC1	
R-0	

15	0
CAUC1	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16-15. RTI Capture Up Counter 1 Register (RTICAUC1) Field Descriptions

Bit	Field	Value	Description
31-0	CAUC1	0-FFFF FFFFh	Capture up counter 1. This register captures the current value of the up counter 1 (RTIUC1) when an event occurs, controlled by the external capture control block. **Note: The RTICAFRC1 register must be read before the RTICAUC1 register is read. This sequence ensures that the value of the RTICAUC1 register is the corresponding value to the RTICAFRC1 register, even if another capture event happens in between the two reads.** A read of this register returns the value of RTIUC1 on a capture event.

16.3.15 RTI Compare 0 Register (RTICOMP0)

The compare 0 register holds the value to be compared with the counters. This register is shown in Figure 16-22 and described in Table 16-16.

Figure 16-22. RTI Compare 0 Register (RTICOMP0) [offset = 50h]

31	16
COMP0	
R/WP-0	

15	0
COMP0	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-16. RTI Compare 0 Register (RTICOMP0) Field Descriptions

Bit	Field	Value	Description
31-0	COMP0	0-FFFF FFFFh	Compare 0. This registers holds a value that is compared with the counter selected in the compare control logic. If RTIFRC0 or RTIFRC1, depending on the counter selected, matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request. A read of this register will return the current compare value. A write to this register (in privileged mode only) will update the compare register with a new compare value.

16.3.16 RTI Update Compare 0 Register (RTIUDCP0)

The update compare 0 register holds the value to be added to the compare register 0 value on a compare match. This register is shown in Figure 16-23 and described in Table 16-17.

Figure 16-23. RTI Update Compare 0 Register (RTIUDCP0) [offset = 54h]

31	16
UDCP0	
R/WP-0	

15	0
UDCP0	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-17. RTI Update Compare 0 Register (RTIUDCP0) Field Descriptions

Bit	Field	Value	Description
31-0	UDCP0	0-FFFF FFFFh	Update compare 0. This register holds a value that is added to the value in the compare 0 (RTICOMP0) register each time a compare matches. This function allows periodic interrupts to be generated without software intervention. A read of this register will return the value to be added to the RTICOMP0 register on the next compare match. A write to this register will provide a new update value.

16.3.17 RTI Compare 1 Register (RTICOMP1)

The compare 1 register holds the value to be compared to the counters. This register is shown in Figure 16-24 and described in Table 16-18.

Figure 16-24. RTI Compare 1 Register (RTICOMP1) [offset = 58h]

31	16
COMP1	
R/WP-0	

15	0
COMP1	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-18. RTI Compare 1 Register (RTICOMP1) Field Descriptions

Bit	Field	Value	Description
31-0	COMP1	0-FFFF FFFFh	Compare 1. This register holds a value that is compared with the counter selected in the compare control logic. If RTIFRC0 or RTIFRC1, depending on the counter selected, matches this compare value, an interrupt is flagged. With this register, it is possible to initiate a DMA request. A read of this register will return the current compare value. A write to this register will update the compare register with a new compare value.

16.3.18 RTI Update Compare 1 Register (RTIUDCP1)

The update compare 1 register holds the value to be added to the compare register 1 value on a compare match. This register is shown in Figure 16-25 and described in Table 16-19.

Figure 16-25. RTI Update Compare 1 Register (RTIUDCP1) [offset = 5Ch]

31	16
UDCP1	
R/WP-0	

15	0
UDCP1	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-19. RTI Update Compare 1 Register (RTIUDCP1) Field Descriptions

Bit	Field	Value	Description
31-0	UDCP1	0-FFFF FFFFh	Update compare 1. This register holds a value that is added to the value in the RTICOMP1 register each time a compare matches. This process allows periodic interrupts to be generated without software intervention. A read of this register will return the value to be added to the RTICOMP1 register on the next compare match. A write to this register will provide a new update value.

16.3.19 RTI Compare 2 Register (RTICOMP2)

The compare 2 register holds the value to be compared to the counters. This register is shown in Figure 16-26 and described in Table 16-20.

Figure 16-26. RTI Compare 2 Register (RTICOMP2) [offset = 60h]

31	16
COMP2	
R/WP-0	

15	0
COMP2	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-20. RTI Compare 2 Register (RTICOMP2) Field Descriptions

Bit	Field	Value	Description
31-0	COMP2	0-FFFF FFFFh	Compare 2. This register holds a value that is compared with the counter selected in the compare control logic. If RTIFRC0 or RTIFRC1, depending on the counter selected, matches this compare value, an interrupt is flagged. With this register, it is possible to initiate a DMA request. A read of this register will return the current compare value. A write to this register (in privileged mode only) will provide a new compare value.

16.3.20 RTI Update Compare 2 Register (RTIUDCP2)

The update compare 2 register holds the value to be added to the compare register 2 value on a compare match. This register is shown in Figure 16-27 and described in Table 16-21.

Figure 16-27. RTI Update Compare 2 Register (RTIUDCP2) [offset = 64h]

31	16
UDCP2	
R/WP-0	

15	0
UDCP2	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-21. RTI Update Compare 2 Register (RTIUDCP2) Field Descriptions

Bit	Field	Value	Description
31-0	UDCP2	0-FFFF FFFFh	Update compare 2. This register holds a value that is added to the value in the RTICOMP2 register each time a compare matches. This process makes it possible to generate periodic interrupts without software intervention. A read of this register will return the value to be added to the RTICOMP2 register on the next compare match. A write to this register will provide a new update value.

16.3.21 RTI Compare 3 Register (RTICOMP3)

The compare 3 register holds the value to be compared to the counters. This register is shown in Figure 16-28 and described in Table 16-22.

Figure 16-28. RTI Compare 3 Register (RTICOMP3) [offset = 68h]

31	16
COMP3	
R/WP-0	

15	0
COMP3	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-22. RTI Compare 3 Register (RTICOMP3) Field Descriptions

Bit	Field	Value	Description
31-0	COMP3	0-FFFF FFFFh	Compare 3. This register holds a value that is compared with the counter selected in the compare control logic. If RTIFRC0 or RTIFRC1, depending on the counter selected, matches this compare value, an interrupt is flagged. With this register, it is possible to initiate a DMA request. A read of this register will return the current compare value. A write to this register will provide a new compare value.

16.3.22 RTI Update Compare 3 Register (RTIUDCP3)

The update compare 3 register holds the value to be added to the compare register 3 value on a compare match. This register is shown in Figure 16-29 and described in Table 16-23.

Figure 16-29. RTI Update Compare 3 Register (RTIUDCP3) [offset = 6Ch]

31	16
UDCP3	
R/WP-0	

15	0
UDCP3	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-23. RTI Update Compare 3 Register (RTIUDCP3) Field Descriptions

Bit	Field	Value	Description
31-0	UDCP3	0-FFFF FFFFh	Update compare 3. This register holds a value that is added to the value in the RTICOMP3 register each time a compare matches. This process makes it possible to generate periodic interrupts without software intervention. A read of this register will return the value to be added to the RTICOMP3 register on the next compare match. A write to this register will provide a new update value.

16.3.23 RTI Timebase Low Compare Register (RTITBLCOMP)

The timebase low compare register holds the value to activate the edge detection circuit. This register is shown in Figure 16-30 and described in Table 16-24.

Figure 16-30. RTI Timebase Low Compare Register (RTITBLCOMP) [offset = 70h]

31	16
TBLCOMP	
R/WP-0	

15	0
TBLCOMP	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-24. RTI Timebase Low Compare Register (RTITBLCOMP) Field Descriptions

Bit	Field	Value	Description
31-0	TBLCOMP	0-FFFF FFFFh	Timebase low compare value. This value determines when the edge detection circuit starts monitoring the NTU signal. It will be compared with RTIUC0. A read of this register will return the current compare value. A write to this register has the following effects: If TBEXT = 0: The compare value is updated. If TBEXT = 1: The compare value is not changed.

16.3.24 RTI Timebase High Compare Register (RTITBHCOMP)

The timebase high compare register holds the value to deactivate the edge detection circuit. This register is shown in Figure 16-31 and described in Table 16-25.

Figure 16-31. RTI Timebase High Compare Register (RTITBHCOMP) [offset = 74h]

31	16
TBHCOMP	
R/WP-0	

15	0
TBHCOMP	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-25. RTI Timebase High Compare Register (RTITBHCOMP) Field Descriptions

Bit	Field	Value	Description
31-0	TBHCOMP	0-FFFF FFFFh	Timebase high compare value. This value determines when the edge detection circuit will stop monitoring the NTU signal. It will be compared with RTIUC0. RTITBHCOMP must be less than RTICPUC0 because RTIUC0 will be reset when RTICPUC0 is reached. Example: The NTU edge detection circuit should be active ± 10 RTICLK cycles around RTICPUC0. • RTICPUC0 = 0050h • RTITBLCOMP = 0046h • RTITBHCOMP = 0009h A read of this register will return the current compare value. A write to this register has the following effects: If TBEXT = 0: The compare value is updated. If TBEXT = 1: The compare value is not changed.

16.3.25 RTI Set Interrupt Enable Register (RTISETINTENA)

This register prevents the necessity of a read-modify-write operation if a particular interrupt should be enabled. This register is shown in Figure 16-32 and described in Table 16-26.

Figure 16-32. RTI Set Interrupt Control Register (RTISETINTENA) [offset = 80h]

31	24
Reserved	
R-0	

23	19	18	17	16
Reserved		SETOVL1INT	SETOVL0INT	SETTBINT
R-0		R/WP-0	R/WP-0	R/WP-0

15	12	11	10	9	8
Reserved		SETDMA3	SETDMA2	SETDMA1	SETDMA0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	4	3	2	1	0
Reserved		SETINT3	SETINT2	SETINT1	SETINT0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-26. RTI Set Interrupt Control Register (RTISETINTENA) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	SETOVL1INT		Set free running counter 1 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
17	SETOVL0INT		Set free running counter 0 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
16	SETTBINT		Set timebase interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SETDMA3		Set compare DMA request 3.
		0	*Read:* DMA request is disabled. *Write:* DMA request is unchanged.
		1	*Read or Write:* DMA request is enabled.
10	SETDMA2		Set compare DMA request 2.
		0	*Read:* DMA request is disabled. *Write:* DMA request is unchanged.
		1	*Read or Write:* DMA request is enabled.
9	SETDMA1		Set compare DMA request 1.
		0	*Read:* DMA request is disabled. *Write:* DMA request is unchanged.
		1	*Read or Write:* DMA request is enabled.

Table 16-26. RTI Set Interrupt Control Register (RTISETINTENA) Field Descriptions (continued)

Bit	Field	Value	Description
8	SETDMA0		Set compare DMA request 0.
		0	*Read:* DMA request is disabled.
			Write: DMA request is unchanged.
		1	*Read or Write:* DMA request is enabled.
7-4	Reserved	0	Reads return 0. Writes have no effect.
3	SETINT3		Set compare interrupt 3.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
2	SETINT2		Set compare interrupt 2.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
1	SETINT1		Set compare interrupt 1.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
0	SETINT0		Set compare interrupt 0.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.

16.3.26 RTI Clear Interrupt Enable Register (RTICLEARINTENA)

This register prevents the necessity of a read-modify-write operation if a particular interrupt should be disabled. This register is shown in Figure 16-33 and described in Table 16-27.

Figure 16-33. RTI Clear Interrupt Control Register (RTICLEARINTENA) [offset = 84h]

31	24
Reserved	
R-0	

23	19	18	17	16
Reserved		CLEAROVL1INT	CLEAROVL0INT	CLEARTBINT
R-0		R/WP-0	R/WP-0	R/WP-0

15	12	11	10	9	8
Reserved		CLEARDMA3	CLEARDMA2	CLEARDMA1	CLEARDMA0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	4	3	2	1	0
Reserved		CLEARINT3	CLEARINT2	CLEARINT1	CLEARINT0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-27. RTI Clear Interrupt Control Register (RTICLEARINTENA) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	CLEAROVL1INT		Clear free running counter 1 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
17	CLEAROVL0INT		Clear free running counter 0 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
16	CLEARTBINT		Clear timebase interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	CLEARDMA3		Clear compare DMA request 3.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.
10	CLEARDMA2		Clear compare DMA request 2.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.

Table 16-27. RTI Clear Interrupt Control Register (RTICLEARINTENA) Field Descriptions (continued)

Bit	Field	Value	Description
9	CLEARDMA1		Clear compare DMA request 1.
		0	*Read:* DMA request is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled.
			Write: DMA request is disabled.
8	CLEARDMA1		Clear compare DMA request 0.
		0	*Read:* DMA request is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled.
			Write: DMA request is disabled.
7-4	Reserved	0	Reads return 0. Writes have no effect.
3	CLEARINT3		Clear compare interrupt 3.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled.
			Write: Interrupt is disabled.
2	CLEARINT2		Clear compare interrupt 2.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled.
			Write: Interrupt is disabled.
1	CLEARINT1		Clear compare interrupt 1.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled.
			Write: Interrupt is disabled.
0	CLEARINT0		Clear compare interrupt 0.
		0	*Read:* Interrupt is disabled.
			Write: Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled.
			Write: Interrupt is disabled.

16.3.27 RTI Interrupt Flag Register (RTIINTFLAG)

The corresponding flags are set at every compare match of the RTIFRCx and RTICOMPx values, whether the interrupt is enabled or not. This register is shown in Figure 16-34 and described in Table 16-28.

Figure 16-34. RTI Interrupt Flag Register (RTIINTFLAG) [offset = 88h]

31	19	18	17	16
Reserved		OVL1INT	OVL0INT	TBINT
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0

15	4	3	2	1	0
Reserved		INT3	INT2	INT1	INT0
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0	R/W1CP-0

LEGEND: R/W = Read/Write; R = Read only; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Table 16-28. RTI Interrupt Flag Register (RTIINTFLAG) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	OVL1INT		Free running counter 1 overflow interrupt flag. This bit determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
17	OVL0INT		Free running counter 0 overflow interrupt flag. This bit determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
16	TBINT		Timebase interrupt flag. This flag is set when the TBEXT bit is cleared by detection of a missing external clock edge. It will not be set by clearing TBEXT by software. It determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
15-4	Reserved	0	Reads return 0. Writes have no effect.
3	INT3		Interrupt flag 3. These bits determine if an interrupt due to a Compare 3 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
2	INT2		Interrupt flag 2. These bits determine if an interrupt due to a Compare 2 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
1	INT1		Interrupt flag 1. These bits determine if an interrupt due to a Compare 1 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.

Table 16-28. RTI Interrupt Flag Register (RTIINTFLAG) Field Descriptions (continued)

Bit	Field	Value	Description
0	INT0	0	Interrupt flag 0. These bits determine if an interrupt due to a Compare 0 match is pending. *Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.

16.3.28 Digital Watchdog Control Register (RTIDWDCTRL)

The software has to write to the DWDCTRL field in order to enable the DWD, as described below. Once enabled, the watchdog can only be disabled by a system reset. The application cannot disable the watchdog. However should the RTICLK source be changed to a source that is unimplemented it will have the same effect as disabling the watchdog. This register is shown in Figure 16-34 and described in Table 16-28.

Figure 16-35. Digital Watchdog Control Register (RTIDWDCTRL) [offset = 90h]

31	16
DWDCTRL	
R/WP-5312h	

15	0
DWDCTRL	
R/WP-ACEDh	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

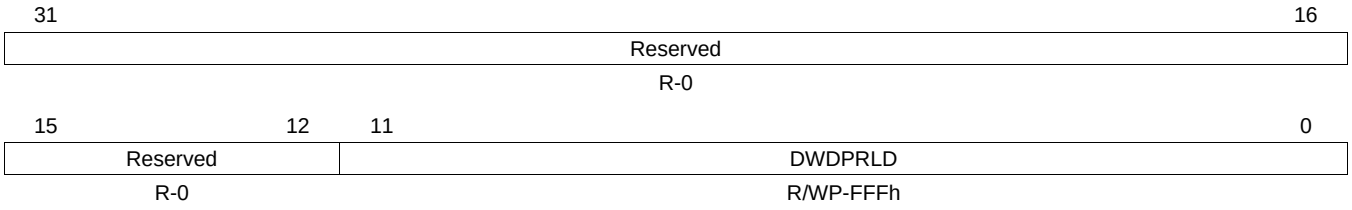
Table 16-29. Digital Watchdog Control Register (RTIDWDCTRL) Field Descriptions

Bit	Field	Value	Description
31-0	DWDCTRL	5312 ACEDh	Digital Watchdog Control. *Read:* DWD counter is disabled. *Write:* State of DWD counter is unchanged (stays enabled or disabled).
		A985 59DAh	*Read:* DWD counter is enabled. *Write:* DWD counter is enabled.
		All other values	*Read:* DWD counter state is unchanged (enabled or disabled). *Write:* State of DWD counter is unchanged (stays enabled or disabled). **Note:** Once the enable value is written, all other future writes are blocked. In other words, once DWD is enabled, it can only be disabled by system reset or power on reset. However should the RTICLK source be changed to a source that is unimplemented it will have the same effect as disabling the watchdog.

16.3.29 Digital Watchdog Preload Register (RTIDWDPRLD)

This register sets the expiration time of the DWD. This register is shown in Figure 16-34 and described in Table 16-28.

Figure 16-36. Digital Watchdog Preload Register (RTIDWDPRLD) [offset = 94h]



LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-30. Digital Watchdog Preload Register (RTIDWDPRLD) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reads return 0 and writes have no effect.
11-0	DWDPRLD	0-FFFh	Digital Watchdog Preload Value. *Read:* The current preload value *Write:* Set the preload value. The DWD preload register can be configured only when the DWD is disabled. Therefore, the application can only configure the DWD preload register before it enables the DWD down counter. The expiration time of the DWD Down Counter can be determined with following equation: texp = (DWDPRLD+1) x 2^{13} / RTICLK1 where: DWDPRLD = 0...4095

16.3.30 Watchdog Status Register (RTIWDSTATUS)

This register records the status of the DWD. The values of the following status bits will not be affected by a soft reset. These bits are cleared by a power up reset, or by a write of '1'. These bits can be used for debug purposes. This register is shown in Figure 16-34 and described in Table 16-28.

Figure 16-37. Watchdog Status Register (RTIWDSTATUS) [offset = 98h]

31	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		DWWD ST	END TIME VIOL	START TIME VIOL	KEY ST	DWD ST	Reserved
R-0		R/W1CP-x	R/W1CP-x	R/W1CP-x	R/W1CP-x	R/W1CP-x	R-0

LEGEND: R/W = Read/Write; R = Read only; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Table 16-31. Watchdog Status Register (RTIWDSTATUS) Field Descriptions

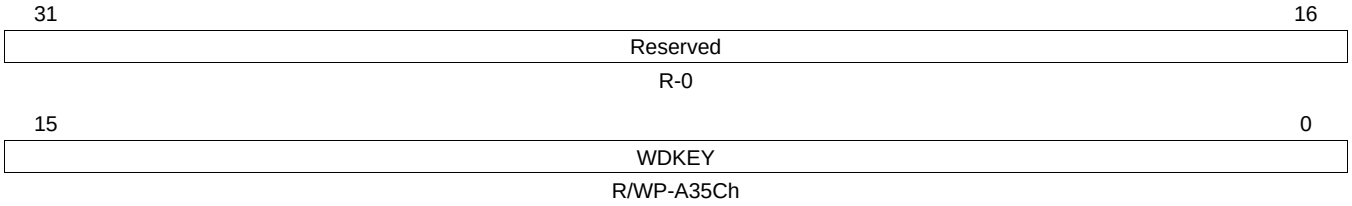
Bit	Field	Value	Description
31-6	Reserved	0	Reads return 0. Writes have no effect.
5	DWWD ST		Windowed Watchdog Status
		0	*Read:* No time-window violation has occurred. *Write:* Leaves the current value unchanged.
		1	*Read:* Time-window violation has occurred. The watchdog has generated either a system reset or a non-maskable interrupt to the CPU in this case. *Write:* Bit is cleared to 0. This will also clear all other status flags in the RTIWDSTATUS register. Clearing of the status flags will deassert the non-maskable interrupt generated due to violation of the DWWD.
4	END TIME VIOL		Windowed Watchdog End Time Violation Status. This bit indicates whether the Watchdog counter expired.
		0	*Read:* No end-time window violation has occurred. *Write:* Leaves the current value unchanged.
		1	*Read*: End-time defined by the windowed watchdog configuration has been violated. *Write:* Bit is cleared to 0.
3	START TIME VIOL		Windowed Watchdog Start Time Violation Status. This bit indicates whether the key is written before the watchdog window opened up.
		0	*Read:* No start-time window violation has occurred. *Write:* Leaves the current value unchanged.
		1	*Read:* Start-time defined by the windowed watchdog configuration has been violated. *Write:* Bit is cleared to 0.
2	KEY ST		Watchdog key status. This bit indicates a reset or NMI generated by a wrong key or key sequence written to the RTIWDKEY register.
		0	*Read:* No wrong key or key-sequence written. *Write:* Bit is unchanged.
		1	*Read:* Wrong key or key-sequence written to RTIWDKEY register. *Write:* Bit is cleared to 0.
1	DWD ST		DWD status. This bit is equivalent to bit END TIME VIOL.
		0	*Read:* No reset or NMI was generated. *Write:* Bit is unchanged.
		1	*Read:* Reset or NMI was generated. *Write:* Bit is cleared to 0.
0	Reserved	0	Reads return 0. Writes have no effect.

16.3.31 RTI Watchdog Key Register (RTIWDKEY)

This register must be written with the correct written key values to serve the watchdog. This register is shown in Figure 16-38 and described in Table 16-32.

NOTE: It has to be taken into account that the write to the RTIWDKEY register takes 3 VCLK cycles.

Figure 16-38. RTI Watchdog Key Register (RTIDWDKEY) [offset = 9Ch]



LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-32. RTI Watchdog Key Register (RTIDWDKEY) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reads return 0 and writes have no effect.
15-0	WDKEY	0-FFFFh	Watchdog key. These bits provide the key sequence location. Reads returns the current WDKEY value. A write of E51Ah followed by A35Ch in two separate write operations defines the key sequence and reloads the DWD. Writing any other value causes a reset or NMI, as shown in Table 16-33. Writing any other value will cause the WDKEY to reset to A35Ch.

Table 16-33. Example of a WDKEY Sequence

Step	Value Written to WDKEY	Result
1	A35Ch	No action
2	A35Ch	No action
3	E51Ah	WDKEY is enabled for reset or NMI by next A35Ch.
4	E51Ah	WDKEY is enabled for reset or NMI by next A35Ch.
5	E51Ah	WDKEY is enabled for reset or NMI by next A35Ch.
6	A35Ch	Watchdog is reset.
7	A35Ch	No action
8	E51Ah	WDKEY is enabled for reset or NMI by next A35Ch.
9	A35Ch	Watchdog is reset.
10	E51Ah	WDKEY is enabled for reset or NMI by next A35Ch.
11	2345h	System reset or NMI; incorrect value written to WDKEY.

16.3.32 RTI Digital Watchdog Down Counter (RTIDWDCNTR)

This register provides the current value of the DWD down counter. This register is shown in Figure 16-39 and described in Table 16-34.

Figure 16-39. RTI Watchdog Down Counter Register (RTIDWDCNTR) [offset = A0h]

31	25	24	16
Reserved		DWDCNTR	
R-0		R-1FFh	

15	0
DWDCNTR	
R-FFFFh	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 16-34. RTI Watchdog Down Counter Register (RTIDWDCNTR) Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reads return 0 and writes have no effect.
24-0	DWDCNTR	0-1FF FFFFh	DWD down counter. Reads return the current counter value.

16.3.33 Digital Windowed Watchdog Reaction Control (RTIWWDRXNCTRL)

This register selects the DWWD reaction if the watchdog is serviced outside the time window. This register is shown in Figure 16-40 and described in Table 16-35.

Figure 16-40. Digital Windowed Watchdog Reaction Control (RTIWWDRXNCTRL) [offset = A4h]

31	16
Reserved	
R-0	

15	4	3	0
Reserved		WWDRXN	
R-0		R/WP-5h	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-35. Digital Windowed Watchdog Reaction Control (RTIWWDRXNCTRL) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reads return 0 and writes have no effect.
3-0	WWDRXN		The DWWD reaction
		5h	The windowed watchdog will cause a reset if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all.
		Ah	The windowed watchdog will generate a non-maskable interrupt to the CPU if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all.
		All other values	The windowed watchdog will cause a reset if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all.
			Note: The DWWD reaction can be selected by the application even when the DWWD counter is already enabled. If a change to the WWDRXN is made before the watchdog service window is opened, then the change in the configuration takes effect immediately. If a change to the WWDRXN is made when the watchdog service window is already open, then the change in configuration takes effect only after the watchdog is serviced.

16.3.34 Digital Windowed Watchdog Window Size Control (RTIWWDSIZECTRL)

This register selects the DWWD window size. This register is shown in Figure 16-41 and described in Table 16-36.

Figure 16-41. Digital Windowed Watchdog Window Size Control (RTIWWDSIZECTRL) [offset = A8h]

31	16
WWDSIZE	
R/WP-0000	

15	0
WWDSIZE	
R/WP-0005h	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-36. Digital Windowed Watchdog Window Size Control (RTIWWDSIZECTRL) Field Descriptions

Bit	Field	Value	Description
31-0	WWDSIZE	0	The DWWD window size
		0000 0005h	100% (The functionality is the same as the standard time-out digital watchdog.)
		0000 0050h	50%
		0000 0500h	25%
		0000 5000h	12.5%
		0005 0000h	6.25%
		All other values	3.125%
			Note: The DWWD window size can be selected by the application even when the DWWD counter is already enabled. If a change to the WWDSIZE is made before the watchdog service window is opened, then the change in the configuration takes effect immediately. If a change to the WWDSIZE is made when the watchdog service window is already open, then the change in configuration takes effect only after the watchdog is serviced.

16.3.35 RTI Compare Interrupt Clear Enable Register (RTIINTCLRENABLE)

When the RTI compare event is configured to generate a DMA request or triggers (all triggered by RTI compare interrupt request flag) to other peripherals, it is often desirable to clear the RTI compare flag automatically so that the requests can be generated repeatedly without any CPU intervention. This register works with the RTI compare clear registers to enable an "auto-clear" of the compare interrupt enable bit after a compare equal event. This register is shown in Figure 16-42 and described in Table 16-37.

Figure 16-42. RTI Compare Interrupt Clear Enable Register (RTIINTCLRENABLE) [offset = ACh]

31 28	27 24	23 20	19 16
Reserved	INTCLRENABLE3	Reserved	INTCLRENABLE2
R-0	R/WP-5h	R-0	R/WP-5h

15 12	11 8	7 4	3 0
Reserved	INTCLRENABLE1	Reserved	INTCLRENABLE0
R-0	R/WP-5h	R-0	R/WP-5h

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-37. RTI Compare Interrupt Clear Enable Register (RTIINTCLRENABLE) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reads return 0. Writes have no effect.
27-24	INTCLRENABLE3	5h	Enables the auto-clear functionality on the compare 3 interrupt. *Read:* Auto-clear for compare 3 interrupt is disabled. *Privileged Write:* Auto-clear for compare 3 interrupt becomes disabled
		All other values	*Read:* Auto-clear for compare 3 interrupt is enabled. *Privileged Write:* Auto-clear for compare 3 interrupt becomes enabled
23-20	Reserved	0	Reads return 0. Writes have no effect.
19-16	INTCLRENABLE2	5h	Enables the auto-clear functionality on the compare 2 interrupt. *Read:* Auto-clear for compare 2interrupt is disabled. *Privileged Write:* Auto-clear for compare 2 interrupt becomes disabled
		All other values	*Read:* Auto-clear for compare 2 interrupt is enabled. *Privileged Write:* Auto-clear for compare 2 interrupt becomes enabled
15-12	Reserved	0	Reads return 0. Writes have no effect.
11-8	INTCLRENABLE1	5h	Enables the auto-clear functionality on the compare 1 interrupt. *Read:* Auto-clear for compare 1 interrupt is disabled. *Privileged Write:* Auto-clear for compare 1 interrupt becomes disabled
		All other values	*Read:* Auto-clear for compare 1 interrupt is enabled. *Privileged Write:* Auto-clear for compare 1 interrupt becomes enabled
7-4	Reserved	0	Reads return 0. Writes have no effect.
3-0	INTCLRENABLE0	5h	Enables the auto-clear functionality on the compare 0 interrupt. *Read:* Auto-clear for compare 0 interrupt is disabled. *Privileged Write:* Auto-clear for compare 0 interrupt becomes disabled
		All other values	*Read:* Auto-clear for compare 0 interrupt is enabled. *Privileged Write:* Auto-clear for compare 0 interrupt becomes enabled

16.3.36 RTI Compare 0 Clear Register (RTICMP0CLR)

This registers holds an initial value which is larger than the value in the RTI Compare 0 register Section 16.3.4. The user needs to choose the value such that the compare clear 0 event occurs before next compare 0 event. If the Free Running Counter matches the compare value, the compare 0 interrupt request flag is cleared and the value in the RTIUDCP0 register Section 16.3.16 is added to this register. This register is shown in Figure 16-43 and described in Table 16-38.

Figure 16-43. RTI Compare 0 Clear Register (RTICMP0CLR) [offset = B0h]

31	16
CMP0CLR	
R/WP-0	

15	0
CMP0CLR	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-38. RTI Compare 0 Clear Register (RTICMP0CLR) Field Descriptions

Bit	Field	Value	Description
31-0	CMP0CLR	0-FFFF FFFFh	Compare 0 clear. This registers holds a compare value. If the Free Running Counter matches the compare value, the compare 0 interrupt request flag is cleared and the value in the RTIUDCP0 register Section 16.3.16 is added to this register. Reads return the current compare clear value. A privileged write to this register updates the compare clear value.

16.3.37 RTI Compare 1 Clear Register (RTICMP1CLR)

This registers holds an initial value which is larger than the value in the RTI Compare 1 register Section 16.3.4. The user needs to choose the value such that the compare clear 1 event occurs before next compare 1 event. If the Free Running Counter matches the compare value, the compare 1 interrupt request flag is cleared and the value in the RTIUDCP1 register Section 16.3.18 is added to this register. This register is shown in Figure 16-44 and described in Table 16-39.

Figure 16-44. RTI Compare 1 Clear Register (RTICMP1CLR) [offset = B4h]

31	16
CMP1CLR	
R/WP-0	

15	0
CMP1CLR	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-39. RTI Compare 1 Clear Register (RTICMP1CLR) Field Descriptions

Bit	Field	Value	Description
31-0	CMP0CLR	0-FFFF FFFFh	Compare 1 clear. This registers holds a compare value. If the Free Running Counter matches the compare value, the compare 1 interrupt request flag is cleared and the value in the RTIUDCP1 register Section 16.3.18 is added to this register. Reads return the current compare clear value. A privileged write to this register updates the compare clear value.

16.3.38 RTI Compare 2 Clear Register (RTICMP2CLR)

This registers holds an initial value which is larger than the value in the RTI Compare 2 register Section 16.3.4. The user needs to choose the value such that the compare clear 2 event occurs before next compare 2 event. If the Free Running Counter matches the compare value, the compare 2 interrupt request flag is cleared and the value in the RTIUDCP2 register Section 16.3.20 is added to this register. This register is shown in Figure 16-45 and described in Table 16-40.

Figure 16-45. RTI Compare 2 Clear Register (RTICMP2CLR) [offset = B8h]

31	16
CMP2CLR	
R/WP-0	

15	0
CMP2CLR	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-40. RTI Compare 2 Clear Register (RTICMP2CLR) Field Descriptions

Bit	Field	Value	Description
31-0	CMP2CLR	0-FFFF FFFFh	Compare 2 clear. This registers holds a compare value. If the Free Running Counter matches the compare value, the compare 2 interrupt request flag is cleared and the value in the RTIUDCP2 register Section 16.3.20 is added to this register. Reads return the current compare clear value. A privileged write to this register updates the compare clear value.

16.3.39 RTI Compare 3 Clear Register (RTICMP3CLR)

This registers holds an initial value which is larger than the value in the RTI Compare 3 register Section 16.3.4. The user needs to choose the value such that the compare clear 3 event occurs before next compare 3 event. If the Free Running Counter matches the compare value, the compare 3 interrupt request flag is cleared and the value in the RTIUDCP3 register Section 16.3.22 is added to this register. This register is shown in Figure 16-46 and described in Table 16-41.

Figure 16-46. RTI Compare 3 Clear Register (RTICMP3CLR) [offset = BCh]

31	16
CMP3CLR	
R/WP-0	

15	0
CMP3CLR	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 16-41. RTI Compare 3 Clear Register (RTICMP3CLR) Field Descriptions

Bit	Field	Value	Description
31-0	CMP3CLR	0-FFFF FFFFh	Compare 3 clear. This registers holds a compare value. If the Free Running Counter matches the compare value, the compare 3 interrupt request flag is cleared and the value in the RTIUDCP3 register Section 16.3.22 is added to this register. Reads return the current compare clear value. A privileged write to this register updates the compare clear value.

General Purpose I/0 (GIO)

This chapter describes the general-purpose input/output (GIO) module. The GIO module provides the family of devices with input/output (I/O) capability. The I/O pins are bidirectional and bit-programmable. The GIO module also supports external interrupt capability.

Topic **Page**

17.1 Overview **2155**
17.2 Quick Start Guide **2155**
17.3 Functional Description of GIO Module **2157**
17.4 Device Modes of Operation **2160**
17.5 MSS_GIO Registers **2161**
17.6 I/O Control Summary **2248**

17.1 Overview

The GIO module offers general-purpose input and output capability. Refer to the device data sheet for identifying the number of GIO ports supported, and the GIO terminals capable of generating an interrupt. Each terminal can be independently configured as input or output and configured as required by the application. The GIO module also supports generation of interrupts whenever a rising edge or falling edge or any toggle is detected on these GIO terminals.

The main features of the GIO module are summarized as follows:

- Allows each GIO terminal to be configured for general-purpose input or output functions
- Supports programmable pull directions on each input GIO terminal
- Supports GIO output in push/pull or open-drain modes
- Allows GIO terminals to be used for generating interrupt requests

17.2 Quick Start Guide

The GIO module comprises two separate components: an input/output (I/O) block and an interrupt generation block. Figure 17-1 and Figure 17-2 show what you should do after reset to configure the GIO module as I/O or for generating interrupts.

In GIO interrupt service routine, you shall read the GIO offset register (GIOOFF1 or GIOOFF2, depending on high-/low-level interrupt) to clear the flag and find the pending interrupt GIO channel.

Figure 17-1. I/O Function Quick Start Flow Chart

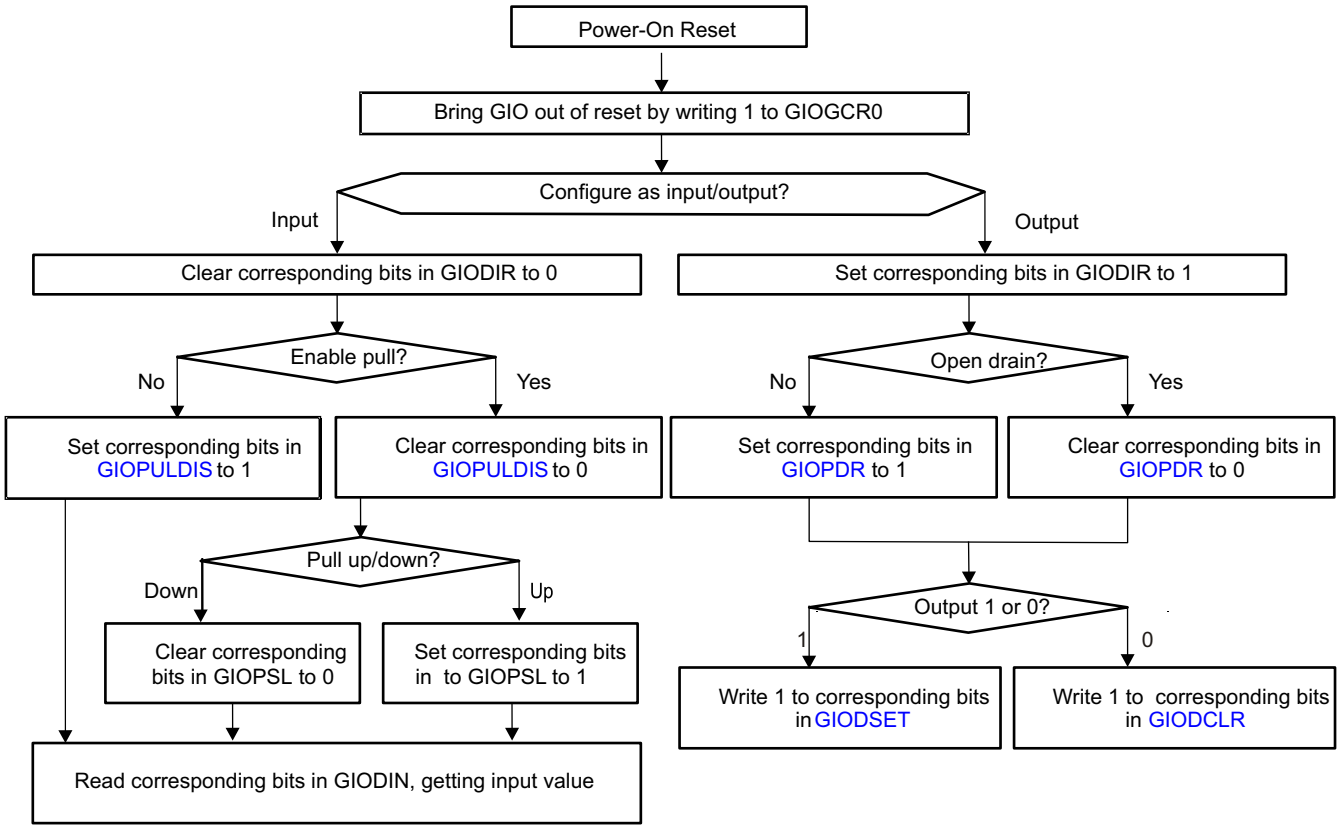
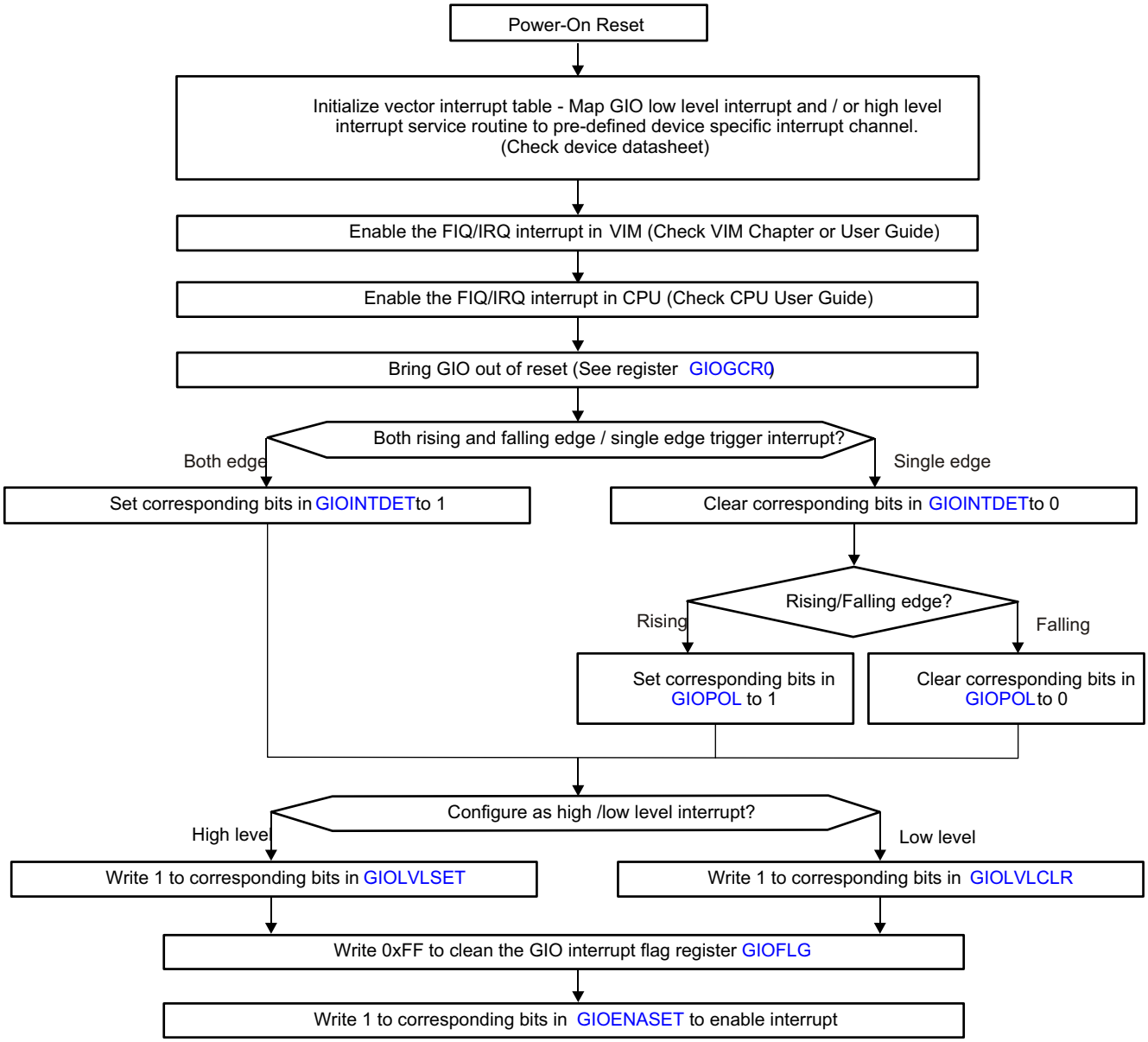


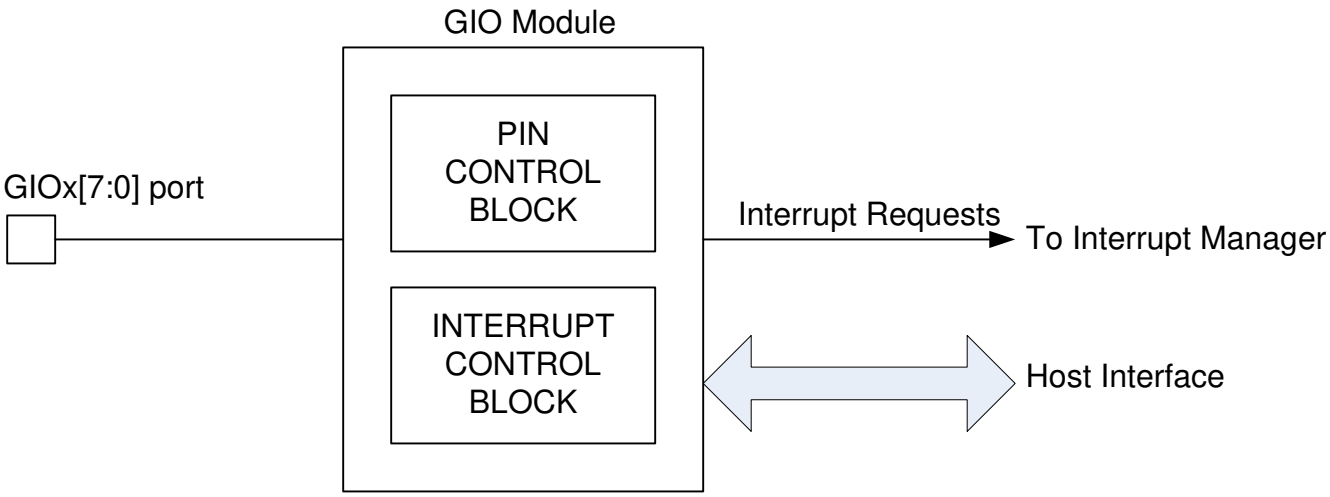
Figure 17-2. Interrupt Generation Function Quick Start Flow Chart



17.3 Functional Description of GIO Module

As shown in Figure 17-3, the GIO module comprises of two separate components: an input/output (I/O) block and an interrupt block.

Figure 17-3. GIO Module Diagram



17.3.1 I/O Functions

The I/O block allows each GIO terminal to be configured for use as a general-purpose input or output in the application. The GIO module supports multiple registers to control the various aspects of the input and output functions. These are described as follows.

- Data direction (GIODIR)

  Configures GIO terminal(s) as input (default) or output through the GIODIRx registers.

- Data input (GIODIN)

  Reflects the logic level on GIO terminals in the GIODINx registers. A high voltage (V_{IH} or greater) applied to the pin causes a high value (1) in the data input register (GIODIN[7:0]). When a low voltage (V_{IL} or less) is applied to the pin, the data input register reads a low value (0). The V_{IH} and V_{IL} values are device specific and can be found in the device datasheet.

- Data output (GIODOUT)

  Configures the logic level to be output on GIO terminal(s) configured as outputs. A low value (0) written to the data output register forces the pin to a low output voltage (V_{OL} or lower). A high value (1) written to the data output register (GIODOUTx) forces the pin to a high output voltage (V_{OH} or higher) if the open drain functionality is disabled (GIOPDRx[7:0]). If open drain functionality is enabled, a high value (1) written to the data output register forces the pin to a high-impedance state (Z).

- Data set (GIODSET)

  Allows logic HIGH to be output on GIO terminal(s) configured as outputs by writing 1's to the required bits in the GIODSETx registers. If open drain functionality is enabled, a high value (1) written to the data output register forces the pin to a high-impedance state (Z). The GIODSETx registers eliminate the need for the application to perform a read-modify-write operation when it needs to set one or more GIO pin(s).

- Data clear (GIODCLR)

  Allows logic LOW to be output on GIO terminal(s) configured as outputs by writing 1s to the required bits in the GIODCLRx registers. The GIODCLRx registers eliminate the need for the application to perform a read-modify-write operation when it needs to clear one or more GIO pin(s).

- Open drain (GIOPDR)

  Open drain functionality is enabled or disabled (default) using the open drain register GIOPDR[7:0] register. If open-drain mode output is enabled on a pin, a high value (1) written to the data output register (GIODOUTx[7:0]) forces the pin to a high impedance state (Z).

- Pull disable (GIOPULDIS)

  Disables the internal pull on GIO terminal(s) configured as inputs by writing to the GIOPULDISx registers.
- Pull select (GIOPSL)

  Selects internal pull down (default) or pull up on GIO terminal(s) configured as inputs by writing to the GIOPULSELx registers.

Refer to the specific device's datasheet to identify the number of GIO ports as well as the input and output functions supported. Some devices may not support the programmable pull controls. In that case, the pull disable and the pull select register controls will not work.

17.3.2 Interrupt Function

The GIO module supports terminals to be configured for generating an interrupt to the host processor through the Vectored Interrupt Manager (VIM). The main functions of the interrupt block are:

- Select the GIO pin(s) that is/are used to generate interrupt(s)

  This is done via the interrupt enable set and clear registers, GIOENASET and GIOENACLR.
- Select the edge on the selected GIO pin(s) that is/are used to generate interrupt(s): rising/falling/both

  Rising or falling edge can be selected via the GIOPOL register. If interrupt is required to be generated on both rising and falling edges, this can be configured via the GIOINTDET register.
- Select the interrupt priority

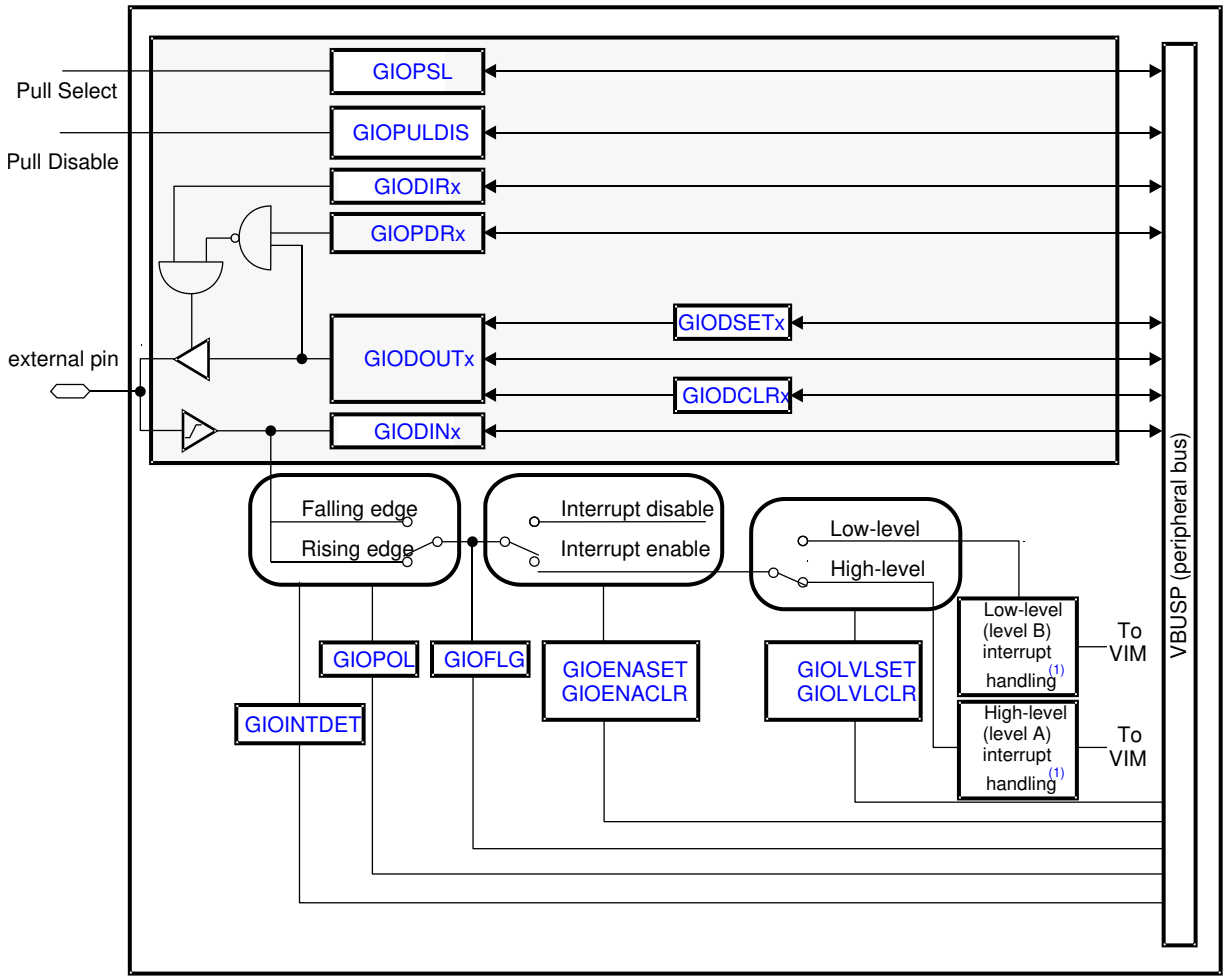
  Low- or high-level interrupt can be selected through the GIOLVLSET and GIOLVLCLR registers.
- Individual interrupt flags are set in the GIOFLG register

Interrupt-capable terminals on GIO ports can be used to handle either general I/O functions or interrupt requests. Each interrupt request can be connected to the VIM at one of two different levels – High (or A) and Low (or B), depending on the VIM channel number. The VIM has an inherent priority scheme so that a request on a lower number channel has a higher priority than a request on a higher number channel. Refer the device datasheet to identify the VIM channel numbers for the GIO level A and level B interrupt requests. Also note that the interrupt priority of level A and level B interrupt handling blocks can be re-programmed in the VIM.

17.3.3 GIO Block Diagram

The GIO block diagram (Figure 17-4) represents the flow of information through a pin. The shaded area corresponds to the I/O block; the unshaded area corresponds to the interrupt block.

Figure 17-4. GIO Block Diagram



(1) A single low-level-interrupt-handling block and a single high-level-interrupt-handling block service all of the interrupt-capable external pins, but only one pin can be serviced by an interrupt block at a time.

17.4 Device Modes of Operation

The GIO module behaves differently in different modes of operation. There are two main modes:

- Emulation mode
- Power-down mode (low-power mode)

17.4.1 Emulation Mode

Emulation mode is used by debugger tools to stop the CPU at breakpoints to read registers.

NOTE: Emulation Mode and Emulation Registers

Emulation mode is a mode of operation of the device and is separate from the GIO emulation registers (GIOEMU1 and GIOEMU2). The contents of these emulation registers are identical to the contents of GIO offset registers (GIOOFF1 and GIOOFF2). Both emulation registers and GIO offset registers are NOT cleared when they are read in emulation mode. GIO offset registers are cleared when they are read in normal mode (other than emulation mode). The emulation registers are NOT cleared when they are read in normal mode. The intention for the emulation registers is that software can use them without clearing the flags.

During emulation mode:

- External interrupts are not captured because the VIM is unable to service interrupts.
- Any register can be read without affecting the state of the system.
- A write to a register still does affect the state of the system.

17.4.2 Power-Down Mode (Low-Power Mode)

In power-down mode, the clock signal to the GIO module is disabled. Thus, there is no switching and the only current draw comes from leakage current. In power-down mode, interrupt pins become level-sensitive rather than edge-sensitive. The polarity bit changes function from falling-edge-triggered to low-level-triggered and rising-edge-triggered to high-level-triggered. A corresponding level on an interrupt pin pulls the module out of low-power mode, if the interrupt is also enabled to wake up the device out of a low-power mode.

17.4.2.1 Module-Level Power Down

The GIO module can be placed into a power down state by disabling the GIO peripheral module via the appropriate bit in the peripheral power down register.

17.4.2.2 Device-Level Power Down

The entire device can be placed in one of the pre-defined low-power modes: doze, snooze, or sleep using the clock source and clock domain disable registers in the system module.

17.5 MSS_GIO Registers

Table 22-194 lists the MSS_GIO registers. All register offset addresses not listed in Table 22-194 should be considered as reserved locations and the register contents should not be modified.

Table 17-1. MSS_GIO Registers

Offset	Acronym	Register Name	Section
0h	GIOGCR	GIOGCR	Section 17.5.1
4h	GIOPWDN	GIOPWDN	Section 17.5.2
8h	GIOINTDET	GIOINTDET	Section 17.5.3
Ch	GIOPOL	GIOPOL	Section 17.5.4
10h	GIOENASET	GIOENASET	Section 17.5.5
14h	GIOENACLR	GIOENACLR	Section 17.5.6
18h	GIOLVLSET	GIOLVLSET	Section 17.5.7
1Ch	GIOLVLCLR	GIOLVLCLR	Section 17.5.8
20h	GIOFLG	GIOFLG	Section 17.5.9
24h	GIOOFFA	GIOOFFA	Section 17.5.10
28h	GIOOFFB	GIOOFFB	Section 17.5.11
2Ch	GIOEMUA	GIOEMUA	Section 17.5.12
30h	GIOEMUB	GIOEMUB	Section 17.5.13
34h	GIODIRA	GIODIRA	Section 17.5.14
38h	GIODINA	GIODINA	Section 17.5.15
3Ch	GIODOUTA	GIODOUTA	Section 17.5.16
40h	GIOSETA	GIOSETA	Section 17.5.17
44h	GIOCLRA	GIOCLRA	Section 17.5.18
48h	GIOPDRA	GIOPDRA	Section 17.5.19
4Ch	GIOPULDISA	GIOPULDISA	Section 17.5.20
50h	GIOPSLA	GIOPSLA	Section 17.5.21
54h	GIODIRB	GIODIRB	Section 17.5.22
58h	GIODINB	GIODINB	Section 17.5.23
5Ch	GIODOUTB	GIODOUTB	Section 17.5.24
60h	GIOSETB	GIOSETB	Section 17.5.25
64h	GIOCLRB	GIOCLRB	Section 17.5.26
68h	GIOPDRB	GIOPDRB	Section 17.5.27
6Ch	GIOPULDISB	GIOPULDISB	Section 17.5.28
70h	GIOPSLB	GIOPSLB	Section 17.5.29
74h	GIODIRC	GIODIRC	Section 17.5.30
78h	GIODINC	GIODINC	Section 17.5.31
7Ch	GIODOUTC	GIODOUTC	Section 17.5.32
80h	GIOSETC	GIOSETC	Section 17.5.33
84h	GIOCLRC	GIOCLRC	Section 17.5.34
88h	GIOPDRC	GIOPDRC	Section 17.5.35
8Ch	GIOPULDISC	GIOPULDISC	Section 17.5.36
90h	GIOPSLC	GIOPSLC	Section 17.5.37
94h	GIODIRD	GIODIRD	Section 17.5.38
98h	GIODIND	GIODIND	Section 17.5.39
9Ch	GIODOUTD	GIODOUTD	Section 17.5.40
A0h	GIOSETD	GIOSETD	Section 17.5.41
A4h	GIOCLRD	GIOCLRD	Section 17.5.42
A8h	GIOPDRD	GIOPDRD	Section 17.5.43
ACh	GIOPULDISD	GIOPULDISD	Section 17.5.44

Table 17-1. MSS_GIO Registers (continued)

Offset	Acronym	Register Name	Section
B0h	GIOPSLD	GIOPSLD	Section 17.5.45
B4h	GIODIRE	GIODIRE	Section 17.5.46
B8h	GIODINE	GIODINE	Section 17.5.47
BCh	GIODOUTE	GIODOUTE	Section 17.5.48
C0h	GIOSETE	GIOSETE	Section 17.5.49
C4h	GIOCLRE	GIOCLRE	Section 17.5.50
C8h	GIOPDRE	GIOPDRE	Section 17.5.51
CCh	GIOPULDISE	GIOPULDISE	Section 17.5.52
D0h	GIOPSLE	GIOPSLE	Section 17.5.53
D4h	GIODIRF	GIODIRF	Section 17.5.54
D8h	GIODINF	GIODINF	Section 17.5.55
DCh	GIODOUTF	GIODOUTF	Section 17.5.56
E0h	GIOSETF	GIOSETF	Section 17.5.57
E4h	GIOCLRF	GIOCLRF	Section 17.5.58
E8h	GIOPDRF	GIOPDRF	Section 17.5.59
ECh	GIOPULDISF	GIOPULDISF	Section 17.5.60
F0h	GIOPSLF	GIOPSLF	Section 17.5.61
F4h	GIODIRG	GIODIRG	Section 17.5.62
F8h	GIODING	GIODING	Section 17.5.63
FCh	GIODOUTG	GIODOUTG	Section 17.5.64
100h	GIOSETG	GIOSETG	Section 17.5.65
104h	GIOCLRG	GIOCLRG	Section 17.5.66
108h	GIOPDRG	GIOPDRG	Section 17.5.67
10Ch	GIOPULDISG	GIOPULDISG	Section 17.5.68
110h	GIOPSLG	GIOPSLG	Section 17.5.69
114h	GIODIRH	GIODIRH	Section 17.5.70
118h	GIODINH	GIODINH	Section 17.5.71
11Ch	GIODOUTH	GIODOUTH	Section 17.5.72
120h	GIOSETH	GIOSETH	Section 17.5.73
124h	GIOCLRH	GIOCLRH	Section 17.5.74
128h	GIOPDRH	GIOPDRH	Section 17.5.75
12Ch	GIOPULDISH	GIOPULDISH	Section 17.5.76
130h	GIOPSLH	GIOPSLH	Section 17.5.77
134h	GIOSRCA	GIOSRCA	Section 17.5.78
138h	GIOSRCB	GIOSRCB	Section 17.5.79
13Ch	GIOSRCC	GIOSRCC	Section 17.5.80
140h	GIOSRCD	GIOSRCD	Section 17.5.81
144h	GIOSRCE	GIOSRCE	Section 17.5.82
148h	GIOSRCF	GIOSRCF	Section 17.5.83
14Ch	GIOSRCG	GIOSRCG	Section 17.5.84
150h	GIOSRCH	GIOSRCH	Section 17.5.85

17.5.1 GIOGCR Register (Offset = 0h) [reset = 0h]

GIOGCR is shown in Figure 17-5 and described in Table 17-2.

Return to the Summary Table.

GIO reset

Figure 17-5. GIOGCR Register

31	30	29	28	27	26	25	24
NU0							
R/W-0h							
23	22	21	20	19	18	17	16
NU0							
R/W-0h							
15	14	13	12	11	10	9	8
NU0							
R/W-0h							
7	6	5	4	3	2	1	0
NU0							RESET
R/W-0h							R/W-0h

Table 17-2. GIOGCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU0	R/W	0h	Reserved
0	RESET	R/W	0h	GIO reset

17.5.2 GIOPWDN Register (Offset = 4h) [reset = 0h]

GIOPWDN is shown in and described in Table 17-3.

Return to the Summary Table.

GIO power down mode register

Table 17-3. GIOPWDN Register Field Descriptions

Bit	Field	Type	Reset	Description
32-1	NU	R/W	0h	Reserved
0	GIOPWDN	R/W	0h	Writing to the GIOPWDN bit is only allowed in privilege mode. Reading of the GIOPWDN bit is allowed in all modes. Privilege mode (write): 0 = Normal operation; clocks enabled to GIO module 1 = Power-down mode User mode (write): Writes have no effect in user mode. User or privilege mode (read): 0 = Normal operation; clocks enabled to GIO module 1 = Power-down mode

17.5.3 GIOINTDET Register (Offset = 8h) [reset = 0h]

GIOINTDET is shown in Figure 17-6 and described in Table 17-4.

Return to the Summary Table.

Interrupt detection select for pins [0:1] GIO[7:0].

Figure 17-6. GIOINTDET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIOINTDET_3								GIOINTDET_2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIOINTDET_1								GIOINTDET_0							
R/W-0h								R/W-0h							

Table 17-4. GIOINTDET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIOINTDET_3	R/W	0h	Interrupt detection select for pins GIOD[7:0].
23-16	GIOINTDET_2	R/W	0h	Interrupt detection select for pins GIOC[7:0].
15-8	GIOINTDET_1	R/W	0h	Interrupt detection select for pins GIOB[7:0].
7-0	GIOINTDET_0	R/W	0h	Interrupt detection select for pins GIOA[7:0].

17.5.4 GIOPOL Register (Offset = Ch) [reset = 0h]

GIOPOL is shown in Figure 17-7 and described in Table 17-5.

Return to the Summary Table.

Interrupt polarity select for pins [0:1] GIO[7:0].

Figure 17-7. GIOPOL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIOPOL_3								GIOPOL_2								GIOPOL_1								GIOPOL_0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 17-5. GIOPOL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIOPOL_3	R/W	0h	Interrupt polarity select for pins GIOD[7:0]
23-16	GIOPOL_2	R/W	0h	Interrupt polarity select for pins GIOC[7:0]
15-8	GIOPOL_1	R/W	0h	Interrupt polarity select for pins GIOB[7:0]
7-0	GIOPOL_0	R/W	0h	Interrupt polarity select for pins GIOA[7:0]

17.5.5 GIOENASET Register (Offset = 10h) [reset = 0h]

GIOENASET is shown in Figure 17-8 and described in Table 17-6.

Return to the Summary Table.

Interrupt enable for pins [0:1] GIO[7:0].

Figure 17-8. GIOENASET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIOENASET_3								GIOENASET_2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIOENASET_1								GIOENASET_0							
R/W-0h								R/W-0h							

Table 17-6. GIOENASET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIOENASET_3	R/W	0h	Interrupt enable for pins GIOD [7:0]
23-16	GIOENASET_2	R/W	0h	Interrupt enable for pins GIOC [7:0]
15-8	GIOENASET_1	R/W	0h	Interrupt enable for pins GIOB [7:0]
7-0	GIOENASET_0	R/W	0h	Interrupt enable for pins GIOA [7:0]

17.5.6 GIOENACLR Register (Offset = 14h) [reset = 0h]

GIOENACLR is shown in Figure 17-9 and described in Table 17-7.

Return to the Summary Table.

Interrupt enable for pins [0:1] GIO[7:0].

Figure 17-9. GIOENACLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIOENACLR_3								GIOENACLR_2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIOENACLR_1								GIOENACLR_0							
R/W-0h								R/W-0h							

Table 17-7. GIOENACLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIOENACLR_3	R/W	0h	Interrupt enable for pins GIOD [7:0]
23-16	GIOENACLR_2	R/W	0h	Interrupt enable for pins GIOC [7:0]
15-8	GIOENACLR_1	R/W	0h	Interrupt enable for pins GIOB [7:0]
7-0	GIOENACLR_0	R/W	0h	Interrupt enable for pins GIOA [7:0]

17.5.7 GIOLVLSET Register (Offset = 18h) [reset = 0h]

GIOLVLSET is shown in Figure 17-10 and described in Table 17-8.

Return to the Summary Table.

GIO high priority interrupt for pins [0:1] GIO[7:0].

Figure 17-10. GIOLVLSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIOLVLSET_3								GIOLVLSET_2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIOLVLSET_1								GIOLVLSET_0							
R/W-0h								R/W-0h							

Table 17-8. GIOLVLSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIOLVLSET_3	R/W	0h	GIO high priority interrupt for pins GIOD[7:0]
23-16	GIOLVLSET_2	R/W	0h	GIO high priority interrupt for pins GIOC[7:0]
15-8	GIOLVLSET_1	R/W	0h	GIO high priority interrupt for pins GIOB[7:0]
7-0	GIOLVLSET_0	R/W	0h	GIO high priority interrupt for pins GIOA[7:0]

17.5.8 GIOLVLCLR Register (Offset = 1Ch) [reset = 0h]

GIOLVLCLR is shown in Figure 17-11 and described in Table 17-9.

Return to the Summary Table.

GIO low priority interrupt for pins [0:1] GIO[7:0].

Figure 17-11. GIOLVLCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GIOLVLCLR_3								GIOLVLCLR_2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIOLVLCLR_1								GIOLVLCLR_0							
R/W-0h								R/W-0h							

Table 17-9. GIOLVLCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIOLVLCLR_3	R/W	0h	GIO low priority interrupt for pins GIOD[7:0]
23-16	GIOLVLCLR_2	R/W	0h	GIO low priority interrupt for pins GIOC[7:0]
15-8	GIOLVLCLR_1	R/W	0h	GIO low priority interrupt for pins GIOB[7:0]
7-0	GIOLVLCLR_0	R/W	0h	GIO low priority interrupt for pins GIOA[7:0]

17.5.9 GIOFLG Register (Offset = 20h) [reset = 0h]

GIOFLG is shown in Figure 17-12 and described in Table 17-10.

Return to the Summary Table.

GIO flag for pins [0:1] GIO[7:0].

Figure 17-12. GIOFLG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIOFLG_3								GIOFLG_2								GIOFLG_1								GIOFLG_0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 17-10. GIOFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	GIOFLG_3	R/W	0h	GIO flag for pins GIOD[7:0]
23-16	GIOFLG_2	R/W	0h	GIO flag for pins GIOC[7:0]
15-8	GIOFLG_1	R/W	0h	GIO flag for pins GIOB[7:0]
7-0	GIOFLG_0	R/W	0h	GIO flag for pins GIOA[7:0]

17.5.10 GIOOFFA Register (Offset = 24h) [reset = 0h]

GIOOFFA is shown in Figure 17-13 and described in Table 17-11.

Return to the Summary Table.

Index bits for currently pending high-priority interrupt Register A

Figure 17-13. GIOOFFA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU1															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1										GIOOFFA					
R/W-0h										R/W-0h					

Table 17-11. GIOOFFA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU1	R/W	0h	Reserved
5-0	GIOOFFA	R/W	0h	Index bits for currently pending high-priority interrupt Register A

17.5.11 GIOOFFB Register (Offset = 28h) [reset = 0h]

GIOOFFB is shown in Figure 17-14 and described in Table 17-12.

Return to the Summary Table.

Index bits for currently pending high-priority interrupt Register B

Figure 17-14. GIOOFFB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2										GIOOFFB					
R/W-0h										R/W-0h					

Table 17-12. GIOOFFB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU2	R/W	0h	Reserved
5-0	GIOOFFB	R/W	0h	Index bits for currently pending high-priority interrupt Register B

17.5.12 GIOEMUA Register (Offset = 2Ch) [reset = 0h]

GIOEMUA is shown in Figure 17-15 and described in Table 17-13.

Return to the Summary Table.

GIO emulation register A

Figure 17-15. GIOEMUA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3										GIOEMUA					
R/W-0h										R/W-0h					

Table 17-13. GIOEMUA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU3	R/W	0h	Reserved
5-0	GIOEMUA	R/W	0h	GIO emulation register A

17.5.13 GIOEMUB Register (Offset = 30h) [reset = 0h]

GIOEMUB is shown in Figure 17-16 and described in Table 17-14.

Return to the Summary Table.

GIO emulation register B

Figure 17-16. GIOEMUB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU4															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU4										GIOEMUB					
R/W-0h										R/W-0h					

Table 17-14. GIOEMUB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU4	R/W	0h	Reserved
5-0	GIOEMUB	R/W	0h	GIO emulation register B

17.5.14 GIODIRA Register (Offset = 34h) [reset = 0h]

GIODIRA is shown in Figure 17-17 and described in Table 17-15.

Return to the Summary Table.

GIO data direction of pins in Port A

Figure 17-17. GIODIRA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU5																								GIODIRA							
R/W-0h																								R/W-0h							

Table 17-15. GIODIRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU5	R/W	0h	Reserved
7-0	GIODIRA	R/W	0h	GIO data direction of pins in Port A

17.5.15 GIODINA Register (Offset = 38h) [reset = 0h]

GIODINA is shown in Figure 17-18 and described in Table 17-16.

Return to the Summary Table.

GIO data input for pins in port A

Figure 17-18. GIODINA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU11																								GIODINA							
R/W-0h																								R/W-0h							

Table 17-16. GIODINA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU11	R/W	0h	Reserved
7-0	GIODINA	R/W	0h	GIO data input for pins in port A

17.5.16 GIODOUTA Register (Offset = 3Ch) [reset = 0h]

GIODOUTA is shown in Figure 17-19 and described in Table 17-17.

Return to the Summary Table.

GIO data output for pins in port A

Figure 17-19. GIODOUTA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU17																								GIODOUTA							
R/W-0h																								R/W-0h							

Table 17-17. GIODOUTA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU17	R/W	0h	Reserved
7-0	GIODOUTA	R/W	0h	GIO data output for pins in port A

17.5.17 GIOSETA Register (Offset = 40h) [reset = 0h]

GIOSETA is shown in Figure 17-20 and described in Table 17-18.

Return to the Summary Table.

GIO data set for port A

Figure 17-20. GIOSETA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU23																								GIODSETA							
R/W-0h																								R/W-0h							

Table 17-18. GIOSETA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU23	R/W	0h	Reserved
7-0	GIODSETA	R/W	0h	GIO data set for port A

17.5.18 GIOCLRA Register (Offset = 44h) [reset = 0h]

GIOCLRA is shown in Figure 17-21 and described in Table 17-19.

Return to the Summary Table.

GIO data clear for port A

Figure 17-21. GIOCLRA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU29																								GIODCLRA							
R/W-0h																								R/W-0h							

Table 17-19. GIOCLRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU29	R/W	0h	Reserved
7-0	GIODCLRA	R/W	0h	GIO data clear for port A

17.5.19 GIOPDRA Register (Offset = 48h) [reset = 0h]

GIOPDRA is shown in Figure 17-22 and described in Table 17-20.

Return to the Summary Table.

GIO open drain for port A

Figure 17-22. GIOPDRA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU35																								GIOPDRA							
R/W-0h																								R/W-0h							

Table 17-20. GIOPDRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU35	R/W	0h	Reserved
7-0	GIOPDRA	R/W	0h	GIO open drain for port A

17.5.20 GIOPULDISA Register (Offset = 4Ch) [reset = 0h]

GIOPULDISA is shown in Figure 17-23 and described in Table 17-21.

Return to the Summary Table.

GIO pul disable for port A

Figure 17-23. GIOPULDISA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								GIOPULDISA							
R/W-0h								R/W-0h							

Table 17-21. GIOPULDISA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved
7-0	GIOPULDISA	R/W	0h	GIO pull disable for port A

17.5.21 GIOPSLA Register (Offset = 50h) [reset = 0h]

GIOPSLA is shown in Figure 17-24 and described in Table 17-22.

Return to the Summary Table.

GIO pul select for port A

Figure 17-24. GIOPSLA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU35																								GIOPSLA							
R/W-0h																								R/W-0h							

Table 17-22. GIOPSLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU35	R/W	0h	Reserved
7-0	GIOPSLA	R/W	0h	GIO pull select for port A

17.5.22 GIODIRB Register (Offset = 54h) [reset = 0h]

GIODIRB is shown in Figure 17-25 and described in Table 17-23.

Return to the Summary Table.

GIO data direction of pins in Port B

Figure 17-25. GIODIRB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU6																								GIODIRB							
R/W-0h																								R/W-0h							

Table 17-23. GIODIRB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU6	R/W	0h	Reserved
7-0	GIODIRB	R/W	0h	GIO data direction of pins in Port B

17.5.23 GIODINB Register (Offset = 58h) [reset = 0h]

GIODINB is shown in Figure 17-26 and described in Table 17-24.

Return to the Summary Table.

GIO data input for pins in port B

Figure 17-26. GIODINB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU12																								GIODINB							
R/W-0h																								R/W-0h							

Table 17-24. GIODINB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU12	R/W	0h	Reserved
7-0	GIODINB	R/W	0h	GIO data input for pins in port B

17.5.24 GIODOUTB Register (Offset = 5Ch) [reset = 0h]

GIODOUTB is shown in Figure 17-27 and described in Table 17-25.

Return to the Summary Table.

GIO data output for pins in port B

Figure 17-27. GIODOUTB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU18																								GIODOUTB							
R/W-0h																								R/W-0h							

Table 17-25. GIODOUTB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU18	R/W	0h	Reserved
7-0	GIODOUTB	R/W	0h	GIO data output for pins in port B

17.5.25 GIOSETB Register (Offset = 60h) [reset = 0h]

GIOSETB is shown in Figure 17-28 and described in Table 17-26.

Return to the Summary Table.

GIO data set for port B

Figure 17-28. GIOSETB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU24																								GIODSETB							
R/W-0h																								R/W-0h							

Table 17-26. GIOSETB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU24	R/W	0h	Reserved
7-0	GIODSETB	R/W	0h	GIO data set for port B

17.5.26 GIOCLRB Register (Offset = 64h) [reset = 0h]

GIOCLRB is shown in Figure 17-29 and described in Table 17-27.

Return to the Summary Table.

GIO data clear for port B

Figure 17-29. GIOCLRB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU30																								GIODCLRB							
R/W-0h																								R/W-0h							

Table 17-27. GIOCLRB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU30	R/W	0h	Reserved
7-0	GIODCLRB	R/W	0h	GIO data clear for port B

17.5.27 GIOPDRB Register (Offset = 68h) [reset = 0h]

GIOPDRB is shown in Figure 17-30 and described in Table 17-28.

Return to the Summary Table.

GIO open drain for port B

Figure 17-30. GIOPDRB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU36																								GIOPDRB							
R/W-0h																								R/W-0h							

Table 17-28. GIOPDRB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOPDRB	R/W	0h	GIO open drain for port B

17.5.28 GIOPULDISB Register (Offset = 6Ch) [reset = 0h]

GIOPULDISB is shown in Figure 17-31 and described in Table 17-29.

Return to the Summary Table.

GIO pul disable for port B

Figure 17-31. GIOPULDISB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU36															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU36								GIOPULDISB							
R/W-0h								R/W-0h							

Table 17-29. GIOPULDISB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOPULDISB	R/W	0h	GIO pull disable for port B

17.5.29 GIOPSLB Register (Offset = 70h) [reset = 0h]

GIOPSLB is shown in Figure 17-32 and described in Table 17-30.

Return to the Summary Table.

GIO pul select for port B

Figure 17-32. GIOPSLB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU36																								GIOPSLB							
R/W-0h																								R/W-0h							

Table 17-30. GIOPSLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOPSLB	R/W	0h	GIO pull select for port B

17.5.30 GIODIRC Register (Offset = 74h) [reset = 0h]

GIODIRC is shown in Figure 17-33 and described in Table 17-31.

Return to the Summary Table.

GIO data direction of pins in Port C

Figure 17-33. GIODIRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU7																								GIODIRC							
R/W-0h																								R/W-0h							

Table 17-31. GIODIRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU7	R/W	0h	Reserved
7-0	GIODIRC	R/W	0h	GIO data direction of pins in Port C

17.5.31 GIODINC Register (Offset = 78h) [reset = 0h]

GIODINC is shown in Figure 17-34 and described in Table 17-32.

Return to the Summary Table.

GIO data input for pins in port C

Figure 17-34. GIODINC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU13																								GIODINC							
R/W-0h																								R/W-0h							

Table 17-32. GIODINC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU13	R/W	0h	Reserved
7-0	GIODINC	R/W	0h	GIO data input for pins in port C

17.5.32 GIODOUTC Register (Offset = 7Ch) [reset = 0h]

GIODOUTC is shown in Figure 17-35 and described in Table 17-33.

Return to the Summary Table.

GIO data output for pins in port C

Figure 17-35. GIODOUTC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU19																								GIODOUTC							
R/W-0h																								R/W-0h							

Table 17-33. GIODOUTC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU19	R/W	0h	Reserved
7-0	GIODOUTC	R/W	0h	GIO data output for pins in port C

17.5.33 GIOSETC Register (Offset = 80h) [reset = 0h]

GIOSETC is shown in Figure 17-36 and described in Table 17-34.

Return to the Summary Table.

GIO data set for port C

Figure 17-36. GIOSETC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU25																								GIODSETC							
R/W-0h																								R/W-0h							

Table 17-34. GIOSETC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU25	R/W	0h	Reserved
7-0	GIODSETC	R/W	0h	GIO data set for port C

17.5.34 GIOCLRC Register (Offset = 84h) [reset = 0h]

GIOCLRC is shown in Figure 17-37 and described in Table 17-35.

Return to the Summary Table.

GIO data clear for port C

Figure 17-37. GIOCLRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU31																								GIODCLRC							
R/W-0h																								R/W-0h							

Table 17-35. GIOCLRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU31	R/W	0h	Reserved
7-0	GIODCLRC	R/W	0h	GIO data clear for port C

17.5.35 GIOPDRC Register (Offset = 88h) [reset = 0h]

GIOPDRC is shown in Figure 17-38 and described in Table 17-36.

Return to the Summary Table.

GIO open drain for port C

Figure 17-38. GIOPDRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU37																								GIOPDRC							
R/W-0h																								R/W-0h							

Table 17-36. GIOPDRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOPDRC	R/W	0h	GIO open drain for port C

17.5.36 GIOPULDISC Register (Offset = 8Ch) [reset = 0h]

GIOPULDISC is shown in Figure 17-39 and described in Table 17-37.

Return to the Summary Table.

GIO pul disable for port C

Figure 17-39. GIOPULDISC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU37															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU37								GIOPULDISC							
R/W-0h								R/W-0h							

Table 17-37. GIOPULDISC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOPULDISC	R/W	0h	GIO pull disable for port C

17.5.37 GIOPSLC Register (Offset = 90h) [reset = 0h]

GIOPSLC is shown in Figure 17-40 and described in Table 17-38.

Return to the Summary Table.

GIO pul select for port C

Figure 17-40. GIOPSLC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU37																								GIOPSLC							
R/W-0h																								R/W-0h							

Table 17-38. GIOPSLC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOPSLC	R/W	0h	GIO pull select for port C

17.5.38 GIODIRD Register (Offset = 94h) [reset = 0h]

GIODIRD is shown in Figure 17-41 and described in Table 17-39.

Return to the Summary Table.

GIO data direction of pins in Port D

Figure 17-41. GIODIRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU8																								GIODIRD							
R/W-0h																								R/W-0h							

Table 17-39. GIODIRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU8	R/W	0h	Reserved
7-0	GIODIRD	R/W	0h	GIO data direction of pins in Port D

17.5.39 GIODIND Register (Offset = 98h) [reset = 0h]

GIODIND is shown in Figure 17-42 and described in Table 17-40.

Return to the Summary Table.

GIO data input for pins in port D

Figure 17-42. GIODIND Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU14																								GIODIND							
R/W-0h																								R/W-0h							

Table 17-40. GIODIND Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU14	R/W	0h	Reserved
7-0	GIODIND	R/W	0h	GIO data input for pins in port D

17.5.40 GIODOUTD Register (Offset = 9Ch) [reset = 0h]

GIODOUTD is shown in Figure 17-43 and described in Table 17-41.

Return to the Summary Table.

GIO data output for pins in port D

Figure 17-43. GIODOUTD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU20																								GIODOUTD							
R/W-0h																								R/W-0h							

Table 17-41. GIODOUTD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU20	R/W	0h	Reserved
7-0	GIODOUTD	R/W	0h	GIO data output for pins in port D

17.5.41 GIOSETD Register (Offset = A0h) [reset = 0h]

GIOSETD is shown in Figure 17-44 and described in Table 17-42.

Return to the Summary Table.

GIO data set for port D

Figure 17-44. GIOSETD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU26																								GIODSETD							
R/W-0h																								R/W-0h							

Table 17-42. GIOSETD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU26	R/W	0h	Reserved
7-0	GIODSETD	R/W	0h	GIO data set for port D

17.5.42 GIOCLRD Register (Offset = A4h) [reset = 0h]

GIOCLRD is shown in Figure 17-45 and described in Table 17-43.

Return to the Summary Table.

GIO data clear for port D

Figure 17-45. GIOCLRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU32																								GIODCLRD							
R/W-0h																								R/W-0h							

Table 17-43. GIOCLRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU32	R/W	0h	Reserved
7-0	GIODCLRD	R/W	0h	GIO data clear for port D

17.5.43 GIOPDRD Register (Offset = A8h) [reset = 0h]

GIOPDRD is shown in Figure 17-46 and described in Table 17-44.

Return to the Summary Table.

GIO open drain for port D

Figure 17-46. GIOPDRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU38																								GIOPDRD							
R/W-0h																								R/W-0h							

Table 17-44. GIOPDRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOPDRD	R/W	0h	GIO open drain for port D

17.5.44 GIOPULDISD Register (Offset = ACh) [reset = 0h]

GIOPULDISD is shown in Figure 17-47 and described in Table 17-45.

Return to the Summary Table.

GIO pul disable for port D

Figure 17-47. GIOPULDISD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU38															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU38								GIOPULDISD							
R/W-0h								R/W-0h							

Table 17-45. GIOPULDISD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOPULDISD	R/W	0h	GIO pull disable for port D

17.5.45 GIOPSLD Register (Offset = B0h) [reset = 0h]

GIOPSLD is shown in Figure 17-48 and described in Table 17-46.

Return to the Summary Table.

GIO pul select for port D

Figure 17-48. GIOPSLD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU38																								GIOPSLD							
R/W-0h																								R/W-0h							

Table 17-46. GIOPSLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOPSLD	R/W	0h	GIO pull select for port D

17.5.46 GIODIRE Register (Offset = B4h) [reset = 0h]

GIODIRE is shown in Figure 17-49 and described in Table 17-47.

Return to the Summary Table.

GIO data direction of pins in Port E

Figure 17-49. GIODIRE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU9																								GIODIRE							
R/W-0h																								R/W-0h							

Table 17-47. GIODIRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU9	R/W	0h	Reserved
7-0	GIODIRE	R/W	0h	GIO data direction of pins in Port E

17.5.47 GIODINE Register (Offset = B8h) [reset = 0h]

GIODINE is shown in Figure 17-50 and described in Table 17-48.

Return to the Summary Table.

GIO data input for pins in port E

Figure 17-50. GIODINE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU15																								GIODINE							
R/W-0h																								R/W-0h							

Table 17-48. GIODINE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU15	R/W	0h	Reserved
7-0	GIODINE	R/W	0h	GIO data input for pins in port E

17.5.48 GIODOUTE Register (Offset = BCh) [reset = 0h]

GIODOUTE is shown in Figure 17-51 and described in Table 17-49.

Return to the Summary Table.

GIO data output for pins in port E

Figure 17-51. GIODOUTE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU21																								GIODOUTE							
R/W-0h																								R/W-0h							

Table 17-49. GIODOUTE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU21	R/W	0h	Reserved
7-0	GIODOUTE	R/W	0h	GIO data output for pins in port E

17.5.49 GIOSETE Register (Offset = C0h) [reset = 0h]

GIOSETE is shown in Figure 17-52 and described in Table 17-50.

Return to the Summary Table.

GIO data set for port E

Figure 17-52. GIOSETE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU27																								GIODSETE							
R/W-0h																								R/W-0h							

Table 17-50. GIOSETE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU27	R/W	0h	Reserved
7-0	GIODSETE	R/W	0h	GIO data set for port E

17.5.50 GIOCLRE Register (Offset = C4h) [reset = 0h]

GIOCLRE is shown in Figure 17-53 and described in Table 17-51.

Return to the Summary Table.

GIO data clear for port E

Figure 17-53. GIOCLRE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU33																								GIODCLRE							
R/W-0h																								R/W-0h							

Table 17-51. GIOCLRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU33	R/W	0h	Reserved
7-0	GIODCLRE	R/W	0h	GIO data clear for port E

17.5.51 GIOPDRE Register (Offset = C8h) [reset = 0h]

GIOPDRE is shown in Figure 17-54 and described in Table 17-52.

Return to the Summary Table.

GIO open drain for port E

Figure 17-54. GIOPDRE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39																								GIOPDRE							
R/W-0h																								R/W-0h							

Table 17-52. GIOPDRE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPDRE	R/W	0h	GIO open drain for port E

17.5.52 GIOPULDISE Register (Offset = CCh) [reset = 0h]

GIOPULDISE is shown in Figure 17-55 and described in Table 17-53.

Return to the Summary Table.

GIO pul disable for port E

Figure 17-55. GIOPULDISE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU39															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39								GIOPULDISE							
R/W-0h								R/W-0h							

Table 17-53. GIOPULDISE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPULDISE	R/W	0h	GIO pull disable for port E

17.5.53 GIOPSLE Register (Offset = D0h) [reset = 0h]

GIOPSLE is shown in Figure 17-56 and described in Table 17-54.

Return to the Summary Table.

GIO pul select for port E

Figure 17-56. GIOPSLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39																								GIOPSLE							
R/W-0h																								R/W-0h							

Table 17-54. GIOPSLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPSLE	R/W	0h	GIO pull select for port E

17.5.54 GIODIRF Register (Offset = D4h) [reset = 0h]

GIODIRF is shown in Figure 17-57 and described in Table 17-55.

Return to the Summary Table.

GIO data direction of pins in Port F

Figure 17-57. GIODIRF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU10																								GIODIRF							
R/W-0h																								R/W-0h							

Table 17-55. GIODIRF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU10	R/W	0h	Reserved
7-0	GIODIRF	R/W	0h	GIO data direction of pins in Port F

17.5.55 GIODINF Register (Offset = D8h) [reset = 0h]

GIODINF is shown in Figure 17-58 and described in Table 17-56.

Return to the Summary Table.

GIO data input for pins in Port F

Figure 17-58. GIODINF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU16																								GIODINF							
R/W-0h																								R/W-0h							

Table 17-56. GIODINF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU16	R/W	0h	Reserved
7-0	GIODINF	R/W	0h	GIO data input for pins in port F

17.5.56 GIODOUTF Register (Offset = DCh) [reset = 0h]

GIODOUTF is shown in Figure 17-59 and described in Table 17-57.

Return to the Summary Table.

GIO data output for pins in Port F

Figure 17-59. GIODOUTF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU22																								GIODOUTF							
R/W-0h																								R/W-0h							

Table 17-57. GIODOUTF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU22	R/W	0h	Reserved
7-0	GIODOUTF	R/W	0h	GIO data output for pins in port F

17.5.57 GIOSETF Register (Offset = E0h) [reset = 0h]

GIOSETF is shown in Figure 17-60 and described in Table 17-58.

Return to the Summary Table.

GIO data set for Port F

Figure 17-60. GIOSETF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU28																								GIODSETF							
R/W-0h																								R/W-0h							

Table 17-58. GIOSETF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU28	R/W	0h	Reserved
7-0	GIODSETF	R/W	0h	GIO data set for port F

17.5.58 GIOCLRF Register (Offset = E4h) [reset = 0h]

GIOCLRF is shown in Figure 17-61 and described in Table 17-59.

Return to the Summary Table.

GIO data clear for Port F

Figure 17-61. GIOCLRF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU34																								GIODCLRF							
R/W-0h																								R/W-0h							

Table 17-59. GIOCLRF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU34	R/W	0h	Reserved
7-0	GIODCLRF	R/W	0h	GIO data clear for port F

17.5.59 GIOPDRF Register (Offset = E8h) [reset = 0h]

GIOPDRF is shown in Figure 17-62 and described in Table 17-60.

Return to the Summary Table.

GIO open drain for Port F

Figure 17-62. GIOPDRF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40																								GIOPDRF							
R/W-0h																								R/W-0h							

Table 17-60. GIOPDRF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPDRF	R/W	0h	GIO open drain for port F

17.5.60 GIOPULDISF Register (Offset = ECh) [reset = 0h]

GIOPULDISF is shown in Figure 17-63 and described in Table 17-61.

Return to the Summary Table.

GIO pul disable for port F

Figure 17-63. GIOPULDISF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU40															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40								GIOPULDISF							
R/W-0h								R/W-0h							

Table 17-61. GIOPULDISF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPULDISF	R/W	0h	GIO pull disable for port F

17.5.61 GIOPSLF Register (Offset = F0h) [reset = 0h]

GIOPSLF is shown in Figure 17-64 and described in Table 17-62.

Return to the Summary Table.

GIO pul select for port F

Figure 17-64. GIOPSLF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40																								GIOPSLF							
R/W-0h																								R/W-0h							

Table 17-62. GIOPSLF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPSLF	R/W	0h	GIO pull select for port F

17.5.62 GIODIRG Register (Offset = F4h) [reset = 0h]

GIODIRG is shown in Figure 17-65 and described in Table 17-63.

Return to the Summary Table.

GIO data direction of pins in Port G

Figure 17-65. GIODIRG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU9																								GIODIRG							
R/W-0h																								R/W-0h							

Table 17-63. GIODIRG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU9	R/W	0h	Reserved
7-0	GIODIRG	R/W	0h	GIO data direction of pins in Port G

17.5.63 GIODING Register (Offset = F8h) [reset = 0h]

GIODING is shown in Figure 17-66 and described in Table 17-64.

Return to the Summary Table.

GIO data input for pins in port G

Figure 17-66. GIODING Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU15																								GIODING							
R/W-0h																								R/W-0h							

Table 17-64. GIODING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU15	R/W	0h	Reserved
7-0	GIODING	R/W	0h	GIO data input for pins in port G

17.5.64 GIODOUTG Register (Offset = FCh) [reset = 0h]

GIODOUTG is shown in Figure 17-67 and described in Table 17-65.

Return to the Summary Table.

GIO data output for pins in port G

Figure 17-67. GIODOUTG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU21																								GIODOUTG							
R/W-0h																								R/W-0h							

Table 17-65. GIODOUTG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU21	R/W	0h	Reserved
7-0	GIODOUTG	R/W	0h	GIO data output for pins in port G

17.5.65 GIOSETG Register (Offset = 100h) [reset = 0h]

GIOSETG is shown in Figure 17-68 and described in Table 17-66.

Return to the Summary Table.

GIO data set for port G

Figure 17-68. GIOSETG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU27																								GIODSETG							
R/W-0h																								R/W-0h							

Table 17-66. GIOSETG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU27	R/W	0h	Reserved
7-0	GIODSETG	R/W	0h	GIO data set for port G

17.5.66 GIOCLRG Register (Offset = 104h) [reset = 0h]

GIOCLRG is shown in Figure 17-69 and described in Table 17-67.

Return to the Summary Table.

GIO data clear for port G

Figure 17-69. GIOCLRG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU33																								GIODCLRG							
R/W-0h																								R/W-0h							

Table 17-67. GIOCLRG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU33	R/W	0h	Reserved
7-0	GIODCLRG	R/W	0h	GIO data clear for port G

17.5.67 GIOPDRG Register (Offset = 108h) [reset = 0h]

GIOPDRG is shown in Figure 17-70 and described in Table 17-68.

Return to the Summary Table.

GIO open drain for port G

Figure 17-70. GIOPDRG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39																								GIOPDRG							
R/W-0h																								R/W-0h							

Table 17-68. GIOPDRG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPDRG	R/W	0h	GIO open drain for port G

17.5.68 GIOPULDISG Register (Offset = 10Ch) [reset = 0h]

GIOPULDISG is shown in Figure 17-71 and described in Table 17-69.

Return to the Summary Table.

GIO pul disable for port G

Figure 17-71. GIOPULDISG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU39															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39								GIOPULDISG							
R/W-0h								R/W-0h							

Table 17-69. GIOPULDISG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPULDISG	R/W	0h	GIO pull disable for port G

17.5.69 GIOPSLG Register (Offset = 110h) [reset = 0h]

GIOPSLG is shown in Figure 17-72 and described in Table 17-70.

Return to the Summary Table.

GIO pul select for port G

Figure 17-72. GIOPSLG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39																								GIOPSLG							
R/W-0h																								R/W-0h							

Table 17-70. GIOPSLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPSLG	R/W	0h	GIO pull select for port G

17.5.70 GIODIRH Register (Offset = 114h) [reset = 0h]

GIODIRH is shown in Figure 17-73 and described in Table 17-71.

Return to the Summary Table.

GIO data direction of pins in Port H

Figure 17-73. GIODIRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU10																								GIODIRH							
R/W-0h																								R/W-0h							

Table 17-71. GIODIRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU10	R/W	0h	Reserved
7-0	GIODIRH	R/W	0h	GIO data direction of pins in Port H

17.5.71 GIODINH Register (Offset = 118h) [reset = 0h]

GIODINH is shown in Figure 17-74 and described in Table 17-72.

Return to the Summary Table.

GIO data input for pins in Port H

Figure 17-74. GIODINH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU16																								GIODINH							
R/W-0h																								R/W-0h							

Table 17-72. GIODINH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU16	R/W	0h	Reserved
7-0	GIODINH	R/W	0h	GIO data input for pins in port H

17.5.72 GIODOUTH Register (Offset = 11Ch) [reset = 0h]

GIODOUTH is shown in Figure 17-75 and described in Table 17-73.

Return to the Summary Table.

GIO data output for pins in Port H

Figure 17-75. GIODOUTH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU22																								GIODOUTH							
R/W-0h																								R/W-0h							

Table 17-73. GIODOUTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU22	R/W	0h	Reserved
7-0	GIODOUTH	R/W	0h	GIO data output for pins in port H

17.5.73 GIOSETH Register (Offset = 120h) [reset = 0h]

GIOSETH is shown in Figure 17-76 and described in Table 17-74.

Return to the Summary Table.

GIO data set for Port H

Figure 17-76. GIOSETH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU28																								GIODSETH							
R/W-0h																								R/W-0h							

Table 17-74. GIOSETH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU28	R/W	0h	Reserved
7-0	GIODSETH	R/W	0h	GIO data set for port H

17.5.74 GIOCLRH Register (Offset = 124h) [reset = 0h]

GIOCLRH is shown in Figure 17-77 and described in Table 17-75.

Return to the Summary Table.

GIO data clear for Port H

Figure 17-77. GIOCLRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU34																								GIODCLRH							
R/W-0h																								R/W-0h							

Table 17-75. GIOCLRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU34	R/W	0h	Reserved
7-0	GIODCLRH	R/W	0h	GIO data clear for port H

17.5.75 GIOPDRH Register (Offset = 128h) [reset = 0h]

GIOPDRH is shown in Figure 17-78 and described in Table 17-76.

Return to the Summary Table.

GIO open drain for Port H

Figure 17-78. GIOPDRH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40																								GIOPDRH							
R/W-0h																								R/W-0h							

Table 17-76. GIOPDRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPDRH	R/W	0h	GIO open drain for port H

17.5.76 GIOPULDISH Register (Offset = 12Ch) [reset = 0h]

GIOPULDISH is shown in Figure 17-79 and described in Table 17-77.

Return to the Summary Table.

GIO pul disable for port H

Figure 17-79. GIOPULDISH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU40															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40								GIOPULDISH							
R/W-0h								R/W-0h							

Table 17-77. GIOPULDISH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPULDISH	R/W	0h	GIO pull disable for port H

17.5.77 GIOPSLH Register (Offset = 130h) [reset = 0h]

GIOPSLH is shown in Figure 17-80 and described in Table 17-78.

Return to the Summary Table.

GIO pul select for port H

Figure 17-80. GIOPSLH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40																								GIOPSLH							
R/W-0h																								R/W-0h							

Table 17-78. GIOPSLH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPSLH	R/W	0h	GIO pull select for port H

17.5.78 GIOSRCA Register (Offset = 134h) [reset = 0h]

GIOSRCA is shown in Figure 17-81 and described in Table 17-79.

Return to the Summary Table.

GIO slew rate select for port A

Figure 17-81. GIOSRCA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU35																								GIOSRCA							
R/W-0h																								R/W-0h							

Table 17-79. GIOSRCA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU35	R/W	0h	Reserved
7-0	GIOSRCA	R/W	0h	GIO slew rate control for port A

17.5.79 GIOSRCB Register (Offset = 138h) [reset = 0h]

GIOSRCB is shown in Figure 17-82 and described in Table 17-80.

Return to the Summary Table.

GIO slew rate select for port B

Figure 17-82. GIOSRCB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU36																								GIOSRCB							
R/W-0h																								R/W-0h							

Table 17-80. GIOSRCB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOSRCB	R/W	0h	GIO slew rate control for port B

17.5.80 GIOSRCC Register (Offset = 13Ch) [reset = 0h]

GIOSRCC is shown in Figure 17-83 and described in Table 17-81.

Return to the Summary Table.

GIO slew rate select for port C

Figure 17-83. GIOSRCC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU37																								GIOSRCC							
R/W-0h																								R/W-0h							

Table 17-81. GIOSRCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOSRCC	R/W	0h	GIO slew rate control for port C

17.5.81 GIOSRCD Register (Offset = 140h) [reset = 0h]

GIOSRCD is shown in Figure 17-84 and described in Table 17-82.

Return to the Summary Table.

GIO slew rate select for port D

Figure 17-84. GIOSRCD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU38																								GIOSRCD							
R/W-0h																								R/W-0h							

Table 17-82. GIOSRCD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOSRCD	R/W	0h	GIO slew rate control for port D

17.5.82 GIOSRCE Register (Offset = 144h) [reset = 0h]

GIOSRCE is shown in Figure 17-85 and described in Table 17-83.

Return to the Summary Table.

GIO slew rate select for port E

Figure 17-85. GIOSRCE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39																								GIOSRCE							
R/W-0h																								R/W-0h							

Table 17-83. GIOSRCE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOSRCE	R/W	0h	GIO slew rate control for port E

17.5.83 GIOSRCF Register (Offset = 148h) [reset = 0h]

GIOSRCF is shown in Figure 17-86 and described in Table 17-84.

Return to the Summary Table.

GIO slew rate select for port F

Figure 17-86. GIOSRCF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40																								GIOSRCF							
R/W-0h																								R/W-0h							

Table 17-84. GIOSRCF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOSRCF	R/W	0h	GIO slew rate control for port F

17.5.84 GIOSRCG Register (Offset = 14Ch) [reset = 0h]

GIOSRCG is shown in Figure 17-87 and described in Table 17-85.

Return to the Summary Table.

GIO slew rate select for port G

Figure 17-87. GIOSRCG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU39																								GIOSRCG							
R/W-0h																								R/W-0h							

Table 17-85. GIOSRCG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOSRCG	R/W	0h	GIO slew rate control for port G

17.5.85 GIOSRCH Register (Offset = 150h) [reset = 0h]

GIOSRCH is shown in Figure 17-88 and described in Table 17-86.

Return to the Summary Table.

GIO slew rate select for port H

Figure 17-88. GIOSRCH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU40																								GIOSRCH							
R/W-0h																								R/W-0h							

Table 17-86. GIOSRCH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOSRCH	R/W	0h	GIO slew rate control for port H

17.6 I/O Control Summary

The behavior of the output buffer and the pull control is summarized in Table 17-87.

Table 17-87. Output Buffer, and Pull Control Behavior for GIO Pins

Module under Reset?	Pin Direction (GIODIR)[1][2]	Open Drain Enable (GIOPDR)[1][3]	Pull Disable (GIOPULDIS)[1][4]	Pull Select (GIOPSL)[1][5]	Pull Control	Output Buffer[6]
Yes	X	X	X	X	Device- and module-specific	Disabled
No	0	X	0	0	Pull down	Disabled
No	0	X	0	1	Pull up	Disabled
No	0	X	1	0	Disabled	Disabled
No	0	X	1	1	Disabled	Disabled
No	1	0	X	X	Disabled	Enabled
No	1	1	X	X	Disabled	Enabled

(1) X = Don't care
(2) GIODIR = 0 for input; 1 for output
(3) See
(4) GIOPULDIS = 0 for enabling pull control; 1 for disabling pull control
(5) GIOPSL= 0 for pull-down functionality; 1 for pull-up functionality
(6) If open drain is enabled, output buffer will be disabled if a high level (1) is being output.

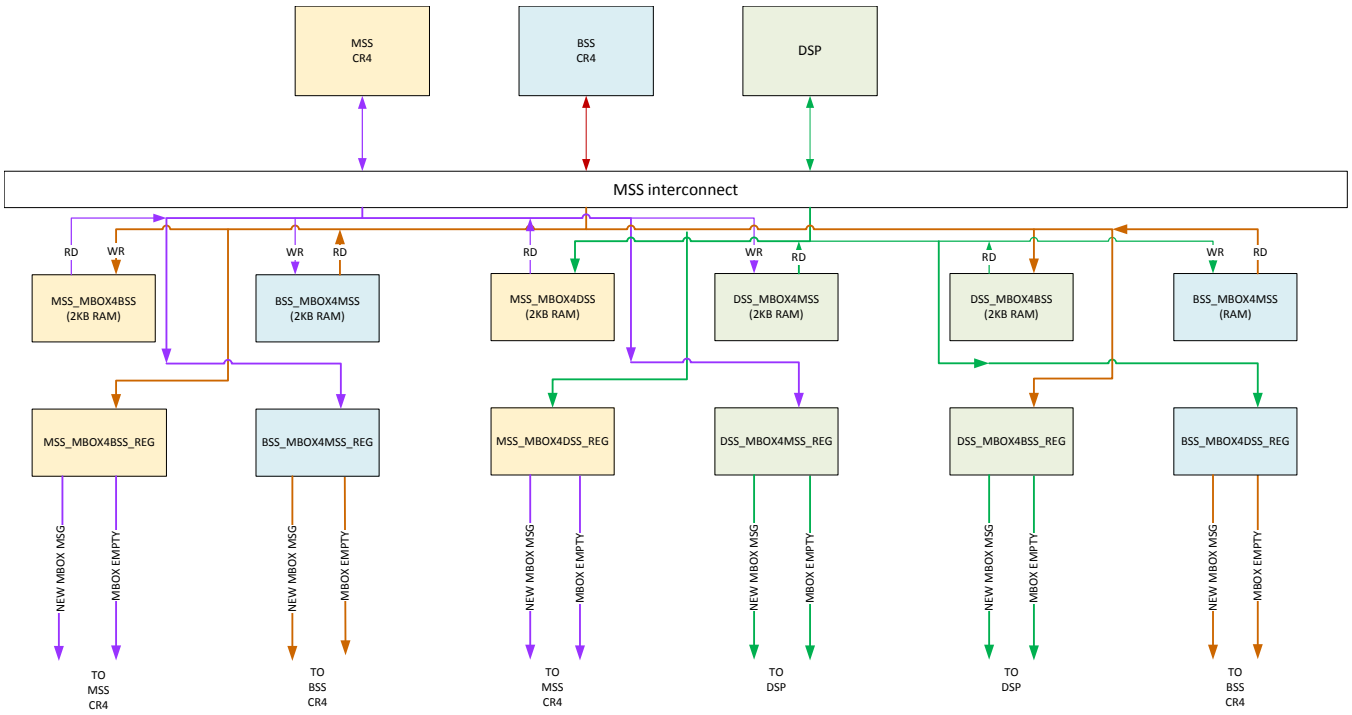
Mailbox

The device provides a mailbox mechanism to asynchronously exchange the messages between any two processors.

Each processor has a mailbox memory space, and registers designated to be used by other processor that wishes to communicate with it.

Each processor has two sets of mailbox memory space and registers, and each set is designated per other processor to communicate with it.

Figure 18-1. Mailbox Block Diagram



Topic **Page**

18.1 Maibox Message Scheme **2250**
18.2 Mailbox Registers **2251**

18.1 Maibox Message Scheme

The processor which wishes to send a message to another processor writes the message to the mailbox memory space, then interrupts the receiver processor. The receiver processor acknowledges the interrupt, then reads the message from the mailbox memory space. The receiver informs the sender that the message is read by an interrupt, which is acknowledged back by the sender. The sender must not initiate another message to the same receiver until the previously initiated mailbox interaction with the same receiver is complete.

Message from sender to receiver:

1. SENDER writes the message in the RECEIVER mailbox RECEIVER_MBOX4SENDER.
2. SENDER triggers an interrupt to RECEIVER by writing 1 to RECEIVER_MBOX4SENDER_REG__INT_TRIG[0]. This generates a mailbox-level interrupt to RECEIVER.
3. RECEIVER clears the interrupt by writing 1 to RECEIVER_MBOX4SENDER_REG__INT_ACK[0].
4. RECEIVER reads the message from the mailbox.
5. RECEIVER triggers an interrupt to SENDER indicating the mailbox is read, by writing 1 to SENDER_MBOX4RECEIVER_REG__INT_TRIG[1].
6. SENDER clears the interrupt by writing 1 to SENDER_MBOX4RECEIVER_REG__INT_ACK[1].

Mailbox message example (message from MSS to DSS):

1. MSS writes the message in the DSS mailbox (DSS_MBOX4MSS).
2. MSS triggers an interrupt to DSS by writing 1 to DSS_MBOX4MSS_REG__INT_TRIG[0]. This generates a mailbox-level interrupt to DSS.
3. DSS clears the interrupt by writing 1 to DSS_MBOX4MSS_REG__INT_ACK[0].
4. DSS reads the message from the mailbox.
5. DSS triggers an interrupt to MSS indicating the mailbox is read, by writing 1 to MSS_MBOX4DSS_REG__INT_TRIG[1].
6. MSS clears the interrupt by writing 1 to MSS_MBOX4DSS_REG__INT_ACK[1].

Mailbox message example (message from DSS to MSS):

1. DSS writes the message in the MSS mailbox (MSS_MBOX4DSS).
2. DSS triggers an interrupt to MSS by writing 1 to MSS_MBOX4DSS_REG__INT_TRIG[0]. This triggers a mailbox-level interrupt to MSS.
3. MSS clears the interrupt by writing 1 to MSS_MBOX4DSS_REG__INT_ACK[0].
4. MSS reads the message from the mailbox
5. MSS triggers an interrupt to DSS indicating the mailbox is read, by writing 1 to DSS_MBOX4MSS_REG__INT_TRIG[1].
6. DSS clears the interrupt by writing 1 to DSS_MBOX4MSS_REG__INT_ACK[1].

18.2 Mailbox Registers

Table 18-1 lists the memory-mapped registers for the Mailbox. All register offset addresses not listed in Table 18-1 should be considered as reserved locations and the register contents should not be modified.

Table 18-1. Mailbox Registers

Offset	Acronym	Register Name	Section
0h	INT_MASK	INT_MASK	Section 18.2.1
8h	INT_MASK_SET	INT_MASK_SET	Section 18.2.2
10h	INT_MASK_CLR	INT_MASK_CLR	Section 18.2.3
18h	INT_STS_CLR	INT_STS_CLR	Section 18.2.4
20h	INT_ACK	INT_ACK	Section 18.2.5
28h	INT_TRIG	INT_TRIG	Section 18.2.6
30h	INT_STS_MASKED	INT_STS_MASKED	Section 18.2.7
38h	INT_STS_RAW	INT_STS_RAW	Section 18.2.8

Complex bit access types are encoded to fit into small table cells. Table 18-2 shows the codes that are used for access types in this section.

Table 18-2. Mailbox Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

18.2.1 INT_MASK Register (Offset = 0h) [reset = 0h]

INT_MASK is shown in Figure 18-2 and described in Table 18-3.

Return to Summary Table.

Figure 18-2. INT_MASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_INT_MASK	MAILBOX_INT_MASK
R/W-0h						R/W-0h	R/W-0h

Table 18-3. INT_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	Global mask register. Setting '1' to any of the bit will mask the interrupt and setting '0' will umask the interrupt. This register maintains the status all the masked interrupts.
1	MAILBOX_ACK_INT_MASK	R/W	0h	Setting '1' to the bit will mask the interrupt and setting '0' will umask the interrupt.
0	MAILBOX_INT_MASK	R/W	0h	Setting '1' to the bit will mask the interrupt and setting '0' will umask the interrupt.

18.2.2 INT_MASK_SET Register (Offset = 8h) [reset = 0h]

INT_MASK_SET is shown in Figure 18-3 and described in Table 18-4.

Return to Summary Table.

Figure 18-3. INT_MASK_SET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_INT_MASK_SET	MAILBOX_INT_MASK_SET
W-0h						W-0h	W-0h

Table 18-4. INT_MASK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	0h	Writing '1' to any of the bit will set the mask corresponding interrupt. Writing '0' have no effect.
1	MAILBOX_ACK_INT_MASK_SET	W	0h	Writing '1' to the bit will set the mask corresponding interrupt. Writing '0' have no effect.
0	MAILBOX_INT_MASK_SET	W	0h	Writing '1' to the bit will set the mask corresponding interrupt. Writing '0' have no effect.

18.2.3 INT_MASK_CLR Register (Offset = 10h) [reset = 0h]

INT_MASK_CLR is shown in Figure 18-4 and described in Table 18-5.

Return to Summary Table.

Figure 18-4. INT_MASK_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_INT_MASK_CLR	MAILBOX_INT_MASK_CLR
W-0h						W-0h	W-0h

Table 18-5. INT_MASK_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	0h	Writing '1' to any of the bit will clear the mask for the corresponding interrupt. Writing '0' have no effect.
1	MAILBOX_ACK_INT_MASK_CLR	W	0h	Writing '1' to the bit will clear the mask for the corresponding interrupt. Writing '0' have no effect.
0	MAILBOX_INT_MASK_CLR	W	0h	Writing '1' to the bit will clear the mask for the corresponding interrupt. Writing '0' have no effect.

18.2.4 INT_STS_CLR Register (Offset = 18h) [reset = 0h]

INT_STS_CLR is shown in Figure 18-5 and described in Table 18-6.

Return to Summary Table.

Figure 18-5. INT_STS_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_INT_STS_CLR	MAILBOX_INT_STS_CLR
W-0h						W-0h	W-0h

Table 18-6. INT_STS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	0h	Writing to this register with any value will clear the status register.
1	MAILBOX_ACK_INT_STS_CLR	W	0h	Writing to this register with any value will clear the status register.
0	MAILBOX_INT_STS_CLR	W	0h	Writing to this register with any value will clear the status register.

18.2.5 INT_ACK Register (Offset = 20h) [reset = 0h]

INT_ACK is shown in Figure 18-6 and described in Table 18-7.

Return to Summary Table.

Figure 18-6. INT_ACK Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_INT_ACK	MAILBOX_INT_ACK
W-0h						W-0h	W-0h

Table 18-7. INT_ACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	0h	This act as individual status bit clear. Writing '1' to any of the bit will clear the corresponding bit in status register. Writing '0' have no effect.
1	MAILBOX_ACK_INT_ACK	W	0h	Writing '1' to will clear the corresponding bit in status register. Writing '0' have no effect.
0	MAILBOX_INT_ACK	W	0h	Writing '1' to will clear the corresponding bit in status register. Writing '0' have no effect.

18.2.6 INT_TRIG Register (Offset = 28h) [reset = 0h]

INT_TRIG is shown in Figure 18-7 and described in Table 18-8.

Return to Summary Table.

Figure 18-7. INT_TRIG Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_TRIG	MAILBOX_INT_TRIG
W-0h						W-0h	W-0h

Table 18-8. INT_TRIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	0h	Reserved
1	MAILBOX_ACK_TRIG	W	0h	writing '1' to this bit indicates to other sytem that read from mailbox is complete.
0	MAILBOX_INT_TRIG	W	0h	Writing '1' to will trigger interrupt and also will set the corresponding bit in status register. Writing '0' have no effect.

18.2.7 INT_STS_MASKED Register (Offset = 30h) [reset = 0h]

INT_STS_MASKED is shown in Figure 18-8 and described in Table 18-9.

Return to Summary Table.

Figure 18-8. INT_STS_MASKED Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_STS_MASKED	MAILBOX_INT_STS_MASKED
R-0h						R-0h	R-0h

Table 18-9. INT_STS_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Gives the current status of triggered interrupt excluding masked interrupt.
1	MAILBOX_ACK_STS_MASKED	R	0h	Gives the current status of mailbox empty interrupt if it is not masked.
0	MAILBOX_INT_STS_MASKED	R	0h	Gives the current status of mailbox full interrupt if it is not masked.

18.2.8 INT_STS_RAW Register (Offset = 38h) [reset = 0h]

INT_STS_RAW is shown in Figure 18-9 and described in Table 18-10.

Return to Summary Table.

Figure 18-9. INT_STS_RAW Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						MAILBOX_ACK_STS_RAW	MAILBOX_INT_STS_RAW
R-0h						R-0h	R-0h

Table 18-10. INT_STS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Gives the current status of triggered interrupt including masked interrupt.
1	MAILBOX_ACK_STS_RAW	R	0h	Gives the current status of mailbox empty interrupt
0	MAILBOX_INT_STS_RAW	R	0h	Gives the current status of mailbox full interrupt

Data Modification Module (DMM)

A Data Modification Module (DMM) gives the ability to write external data into the device memory.

This chapter describes the functionality of the Data Modification Module (DMM), which provides the capability to modify data in the entire 4 GB address space of the device from an external peripheral, with minimal interruption of the application. A DMM gives the ability to write external data into the device memory.

Topic **Page**

19.1 Overview **2261**
19.2 Module Operation **2262**
19.3 Control Registers **2267**

19.1 Overview

19.1.1 Features

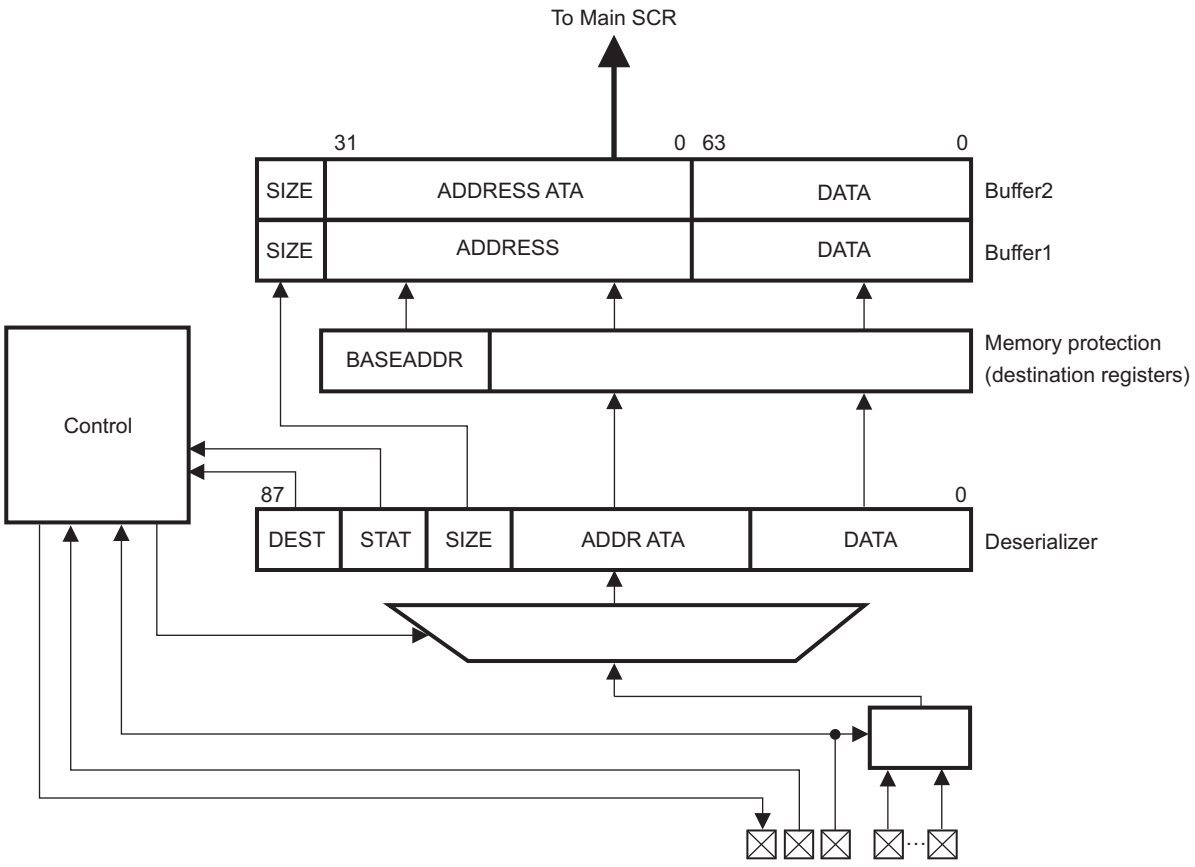
The DMM module has the following features:

- Acts as a bus master, thus enabling direct writes to the 4GB address space without CPU intervention
- Writes to memory locations specified in the received packet (leverages packets defined by trace mode of the RAM trace port (RTP) module
- Writes received data to consecutive addresses, which are specified by the DMM module (leverages packets defined by direct data mode of RTP module)
- Configurable port width (1, 2, 4, 8, 16 pins)
- Up to 65 Mbit/s pin data rate
- Unused pins configurable as GIO pins

19.1.2 Block Diagram

Figure 19-1 shows the block diagram for the DMM.

Figure 19-1. DMM Block Diagram



19.2 Module Operation

The DMM receives data over the DMM pins from external systems and writes the received data directly to the base address programmed in the module plus offset address given in the packet or into a buffer specified by start address and length. It leverages the protocol defined by the RAM Trace Port (RTP) module to have a common interface definition for external systems. It can also be used to connect an RTP and DMM module together for fast processor intercommunication.

The DMM module provides two modes of operation:

- **Trace Mode:** In this mode, the DMM writes the received data directly to an address that is calculated from the base address programmed into the destination register (Section 19.3.12; Section 19.3.14) plus the offset address contained in the received packet. An interrupt can be generated when data is written the lowest address of a programmed region. This capability enables the sender to raise an interrupt at the receiver while sending specific information.
- **Direct Data Mode:** In this mode, the DMM writes the received data into an address range of the 4GB address space. The buffer start address (Section 19.3.8) and blocksize (Section 19.3.9) is programmable in the DMM module. When the buffer reaches its end address, the buffer pointer wraps around and points to the beginning of the buffer again. The EO_BUFF flag (Section 19.3.5) will be set and if enabled, an interrupt will be generated to indicate a buffer-full condition. Another interrupt, can be configured to indicate different buffer fill levels. This can be accomplished by programming a certain fill level into the DMMINTPT register (Section 19.3.11). The PROG_BUFF flag (Section 19.3.5) indicates that this level has been reached.

Data will be captured by the input buffer and moved to the appropriate bit field in the deseralizer. When the deseralizer is completely full, the data will be moved to the output buffer register. A two-level buffer is implemented to avoid overflow conditions if the internal bus is occupied by other transactions. In addition the $\overline{\text{DMMENA}}$ signal can be used to signal the external hardware that an overflow might occur if more data is sent. The automatic generation of the $\overline{\text{DMMENA}}$ signal can be configured by setting the ENAFUNC bit (Section 19.3.16). While the $\overline{\text{DMMENA}}$ signal is active, the DMM module will not receive any new data.

The DMM is a bus master and forwards the received data to the bus system. The write operation will be minimally intrusive to the program flow, because the CPU/DMA access will only be blocked if the CPU/DMA accesses the same resource as the DMM.

To prevent an external system from overwriting critical data in the memory while configured in Trace Mode, a memory protection mechanism is implemented via a programmable start address and block size of a region. A maximum of four destinations with two regions each are supported.

For proper operation, at least DMMCLK, DMMSYNC and DMMDATA[0] need to be programmed in functional mode (Section 19.3.16). If a large amount of data should be transmitted in a short time, more data pins should be used in functional mode. The module supports 1, 2, 4, 8, or 16-pin configurations.

The module can be configured to handle a free running clock provided on DMMCLK (Section 19.3.1). Clock pulses between two DMMSYNC pulses that exceed the number of valid clock pulses for a packet will be ignored.

19.2.1 Data Format

Below is a description of the packet and frame format.

19.2.1.1 Clocking Scheme

The DMM supports both continuous and noncontinuous clocking. The clock received on DMMCLK in the continuous clocking scheme is a free-running clock. In noncontinuous clocking scheme, the clock will stop after each packet and will start with the reception of a DMMSYNC signal.

19.2.1.2 Trace Mode Packet

Figure 19-2 illustrates the trace mode packet format. One packet consists of 2 bits (DEST) denoting the destination in which the data is stored, 2 status bits (STAT), the 2-bit SIZE of the data, the 18-bit address of where the data should be written to, and a variable data field.

The DEST bits (Table 19-1) will be used to determine which destination register applies to the transmitted data and the received address determines if the packet falls into a valid region of the destination area. If the address is valid, the base address, programmed in one of the destination registers (Section 19.3.12; Section 19.3.14) of this particular region will be applied to create the complete 32-bit address for the destination. The DMM module only takes action on a "11" setting of the STAT bits (Table 19-2). This signals that an overflow in the transmitting hardware module has occurred. If this is the case the SRC_OVF flag (Section 19.3.5) will be set and the received data will be written to the address specified in the packet. The size information of the data transmitted in the packet is denoted in the SIZE bits (Table 19-3) of the packet. Depending on the SIZE information, the module expects to receive only this amount of data.

Figure 19-2. Trace Mode Packet Format

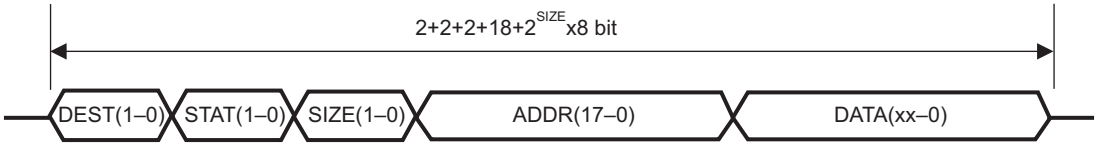


Table 19-1 through Table 19-3 illustrate the encoding of packet format in trace mode.

Table 19-1. Encoding of Destination Bits in Trace Mode Packet Format

DEST[1:0]	Destination
00	Dest 0
01	Dest 1
10	Dest 2
11	Dest 3

Table 19-2. Encoding of Status Bits in Trace Mode Packet Format

STAT[1:0]	Status
00	don't care
01	don't care
10	don't care
11	overflow

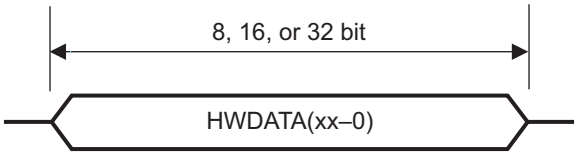
Table 19-3. Encoding of Write Size in Packet Format

SIZE[1:0]	Write Size
00	8 bit
01	16 bit
10	32 bit
11	64 bit

19.2.1.3 Direct Data Mode Packet

Figure 19-3 illustrates the direct data mode packet format.

Figure 19-3. Direct Data Mode Packet Format



The packet consists only of data bits and no header information. It can be 8-, 16- or 32-bit wide. A variable packet width is not supported because the DMM module will check the number of incoming bits (DMMCLK cycles) for error detection. The DMM will write the received data to the destination once the programmed number of bits has been received.

If the programmed word width does not correspond to the received data, the following actions will be taken:

- If the received data is greater than the programmed width, only the configured number of bits are transferred into the RAM buffer, the additional bits are discarded.
- If the received number of bits is smaller than the programmed width, no data will be written to the buffer, because a new DMMSYNC signal has been received before the expected number of bits.

19.2.2 Data Port

The packet will be received in several subpackets, depending on the width of the external data bus (DMMDATA[y:0]) and the amount of data to be transmitted. Table 19-4 illustrates the number of clock cycles required for a complete packet.

Table 19-4. Number of Clock Cycles per Packet

Port Width/ Pins	Write Size in Bits			
	8	16	32	64
1	32	40	56	88
2	16	20	28	44
4	8	10	14	22
8	4	5	7	11
16	2	3	4	6

The user can program the port width in the DMMPC0 register (Section 19.3.16). This feature allows pins that are not used for DMM functionality to be used as GIO pins. Only the pins shown in Table 19-5 can be used for a desired port width.

Table 19-5. Pins Used for Data Communication

Port Width	Pins Used
1	DMMDATA[0]
2	DMMDATA[1:0]
4	DMMDATA[3:0]
8	DMMDATA[7:0]
16	DMMDATA[15:0]

NOTE: If pins other than the ones specified in Table 19-5 are programmed as functional pins for a desired port width, the received data will be corrupted and will not be transferred to the deserializer.

NOTE: If DMMCLK or DMMSYNC are programmed as nonfunctional pins, functional operation will not occur.

19.2.2.1 Signal Description

DMMSYNC	This signal has to be provided by external hardware. It signals the start of a new packet. It has to be active (high) for one full DMMCLK cycle, starting with the rising edge of DMMCLK. If the DMMSYNC pulse is longer than a single DMMCLK cycle and two falling edges of DMMCLK see a high pulse on DMMSYNC, the module will treat the second DMMSYNC pulse as the start of a packet and will flag a PACKET_ERR_INT (Section 19.3.5).
DMMCLK	The clock is externally generated and can be suspended between two packets. For this feature, CONTCLK must be set to 0 (Section 19.3.1). If the clock is not stopped between two packets, CONTCLK must be set to 1. Data will be latched on the falling edge of the DMMCLK signal.
$\overline{\text{DMMENA}}$	This signal is pulled high if no new data should be received via the data pins, because of a potential overflow situation.
DMMDATA[15:0]	These pins receive the packet information transmitted by the external hardware. Data is latched on the falling edge of DMMCLK.

Figure 19-4 shows an example of multiple packets received during trace mode, in noncontinuous clock configuration.

Figure 19-4. Packet Sync Signal Example

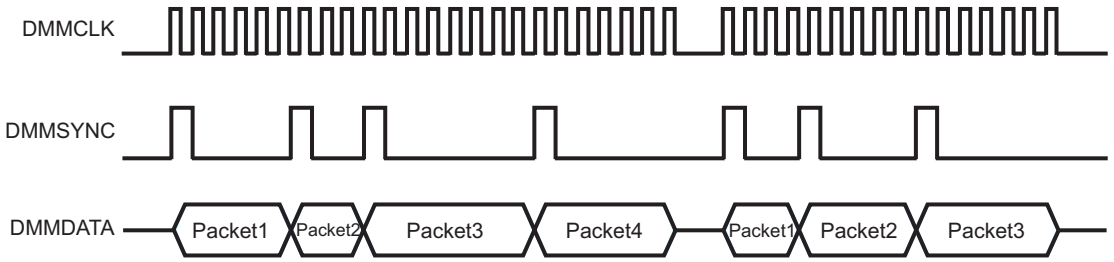
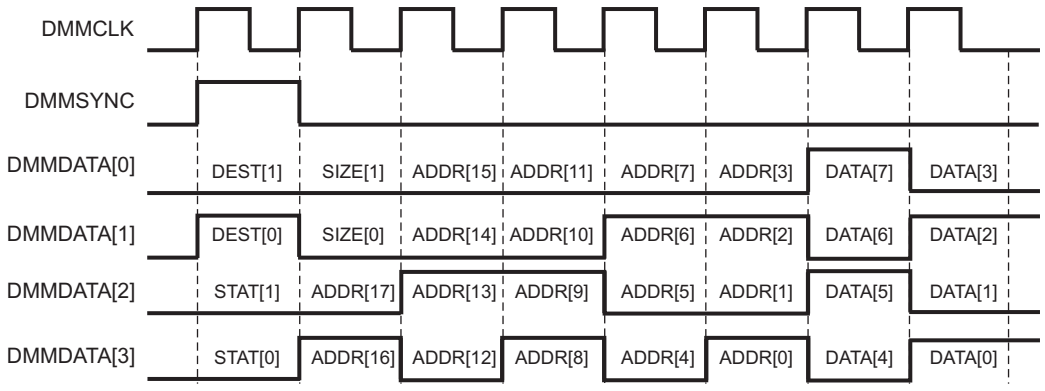


Figure 19-5 shows an example of a 4-bit data port with 8-bit receive data (A5h) to be written into DEST1 (address 0001 2345h) on a trace mode packet.

Figure 19-5. Example Single Packet Transmission



19.2.3 Error Handling

The module will generate two different kind of errors. Once an error condition is recognized, an interrupt will be generated if enabled.

19.2.3.1 Overflow Error

This error is signaled when the module has received new data before the previous data was written to the destination address. If the internal buffers are full, the DMMENA signal will go high. If the sending module does not evaluate the DMMENA signal and keeps on sending new frames, the data that was previously received might be overwritten, thus resulting in setting the BUFF_OVF flag (Section 19.3.5).

19.2.3.2 Packet Error

Noncontinuous Clock Mode

The size of the incoming packet is defined by the SIZE information of a trace mode packet or the programmed size of a direct data mode packet. If too many or less than the number of bits are received before the next sync signal, the PACKET_ERR_INT flag will be set (Section 19.3.5). In case of receiving a DMMCLK signal without a corresponding DMMSYNC signal, a packet error will also be generated.

Continuous Clock Mode

If less than the expected number of bits are received, the PACKET_ERR_INT flag will be set (Section 19.3.5) when the next DMMSYNC signal is received. Packets with more than the expected number of bits cannot be detected.

The check for packet error is done only after the detection of the first DMMSYNC signal after the DMM is turned on or comes out of suspend mode (with COS = 0; Section 19.3.1), that is, before the reception of first DMMSYNC, the toggling of DMMCLK would be ignored.

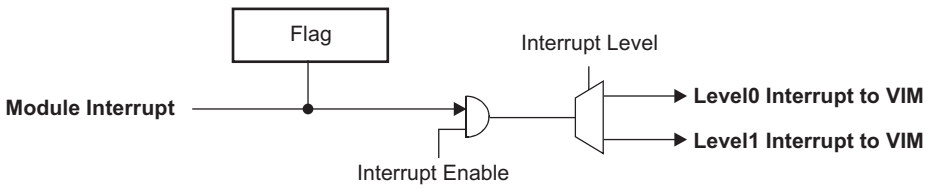
19.2.3.3 Bus Error

If an error occurs on the microcontroller internal bus system while transferring the data from the DMM to the destination, the BUSERROR flag will be set.

19.2.4 Interrupts

The module provides different interrupts. These can be programmed to different interrupt levels independently using DMMINTLVL (Section 19.3.4).

Figure 19-6. Interrupt Structure



Interrupts can be divided into error interrupts and functional interrupts. The error handling is described in Section 19.2.3. Functional interrupts depend on the mode (Trace Mode, Direct Data Mode) the DMM module is used in.

Trace Mode: An interrupt can be enabled whenever an access to the lowest address of a defined region is performed. This address is the starting address programmed in the DMMDESTxREGy register. An interrupt for each of the region can be generated by setting the individual interrupt enable bits.

Direct Data Mode: There are two interrupts that can be individually controlled. One is generated when the buffer pointer reaches the end of the defined buffer and wraps around (EO_BUFF; Section 19.3.2). The other one is generated when the buffer pointer matches the programmed interrupt threshold (PROG_BUFF; Section 19.3.2). The buffer pointer points to the next address to be written, therefor there are (interrupt threshold - 1) values stored in the buffer. The interrupt threshold can be programmed in the DMMINTPT register (Section 19.3.11).

19.3 Control Registers

This section describes the DMM registers. The registers support 8, 16, and 32-bit writes. The offset is relative to the associated peripheral select. Table 19-6 provides a summary of the registers and their bits. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

Table 19-6. DMM Registers

Offset	Acronym	Register Description	Section
0h	DMMGLBCTRL	DMM Global Control Register	Section 19.3.1
4h	DMMINTSET	DMM Interrupt Set Register	Section 19.3.2
8h	DMMINTCLR	DMM Interrupt Clear Register	Section 19.3.3
0Ch	DMMINTLVL	DMM Interrupt Level Register	Section 19.3.4
10h	DMMINTFLG	DMM Interrupt Flag Register	Section 19.3.5
14h	DMMOFF1	DMM Interrupt Offset 1 Register	Section 19.3.6
18h	DMMOFF2	DMM Interrupt Offset 2 Register	Section 19.3.7
1Ch	DMMDDMDEST	DMM Direct Data Mode Destination Register	Section 19.3.8
20h	DMMDDMBL	DMM Direct Data Mode Blocksize Register	Section 19.3.9
24h	DMMDDMPT	DMM Direct Data Mode Pointer Register	Section 19.3.10
28h	DMMINTPT	DMM Direct Data Mode Interrupt Pointer Register	Section 19.3.11
2Ch, 3Ch, 4Ch, 5Ch	DMMDESTxREG1	DMM Destination x Region 1	Section 19.3.12
30h, 40h, 50h, 60h	DMMDESTxBL1	DMM Destination x Blocksize 1	Section 19.3.13
34h, 44h, 54h, 64h	DMMDESTxREG2	DMM Destination x Region 2	Section 19.3.14
38h, 48h, 58h, 68h	DMMDESTxBL2	DMM Destination x Blocksize 2	Section 19.3.15
6Ch	DMMPC0	DMM Pin Control 0	Section 19.3.16
70h	DMMPC1	DMM Pin Control 1	Section 19.3.17
74h	DMMPC2	DMM Pin Control 2	Section 19.3.18
78h	DMMPC3	DMM Pin Control 3	Section 19.3.19
7Ch	DMMPC4	DMM Pin Control 4	Section 19.3.20
80h	DMMPC5	DMM Pin Control 5	Section 19.3.21
84h	DMMPC6	DMM Pin Control 6	Section 19.3.22
88h	DMMPC7	DMM Pin Control 7	Section 19.3.23
8Ch	DMMPC8	DMM Pin Control 8	Section 19.3.24

19.3.1 DMM Global Control Register (DMMGLBCTRL)

With this register the basic operation of the module is selected.

Figure 19-7. DMM Global Control Register (DMMGLBCTRL) [offset = 00h]

31	25	24
Reserved		BUSY
R-0		R-0

23	19	18	17	16
Reserved		CONTCLK	COS	RESET
R-0		R/WP-0	R/WP-0	R/WP-0

15	11	10	9	8
Reserved		DDM_WIDTH		TM_DDM
R-0		R/WP-0		R/WP-0

7	4	3	0
Reserved		ON/OFF	
R-0		R/WP-5h	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-7. DMM Global Control Register (DMMGLBCTRL) Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reads returns 0. Writes have no effect.
24	BUSY		Busy indicator.
		0	The DMM does not currently receive data and has no data in its internal buffers, which needs to be transferred.
		1	The module is currently receiving data, or has data in its internal buffers
23-19	Reserved	0	Reads returns 0. Writes have no effect.
18	CONTCLK		Continuous DMMCLK input.
			User and privilege mode read, privilege mode write:
		0	DMMCLK is expected to be suspended between two packets.
		1	DMMCLK is expected to be free running between packets.
17	COS		Continue on suspend. Influences behavior of module while in debug mode. In all cases the corresponding interrupt will be set.
			User and privilege mode (read):
		0	Packets will not be received during debug mode. Before entering debug mode, the ongoing reception of a packet will be finished and the value will be written to the destination.
		1	Continue receiving packets and update destination, while in debug mode
			Privilege mode (write):
		0	Disable data reception while in debug mode
		1	Enable data reception while in debug mode
16	RESET		Reset. This bit resets the state machine and the registers to its reset value, except the RESET bit itself. It must be cleared by writing to it.
			User and privilege mode (read):
		0	No reset of DMM module
		1	Reset of DMM module
			Privilege mode (write):
		0	No reset of DMM module
		1	Reset DMM module to its reset state
15-11	Reserved	0	Reads returns 0. Writes have no effect.

Table 19-7. DMM Global Control Register (DMMGLBCTRL) Field Descriptions (continued)

Bit	Field	Value	Description
10-9	DDM_WIDTH		Packet Width in direct data mode.
			User and privilege mode read and privilege mode write operation:
		Bit Encoding	Transfer Size
		0	8 bit
		1h	16 bit
		2h	32 bit
		3h	Reserved
8	TM_DMM		Packet Format.
			User and privilege mode (read):
		0	The DMM module assumes packets in trace mode definition
		1	The DMM module assumes packets in direct data mode definition
			Privilege mode (write):
		0	Enable trace mode
		1	Enable direct data mode
7-4	Reserved	0	Reads returns 0. Writes have no effect.
3-0	ON/OFF		Switch module on or off
			User and privilege mode (read):
		All other	The DMM module does not receive data
		Ah	The DMM module receives data and writes it to the buffer
			Privilege mode (write):
		All other	Disable receive/write operations. Packets in reception, will still be finished
		Ah	Enable receive/write operations. Packets will be received 1 HCLK cycle after enabling the module

NOTE: It is recommended to write 5h to ON/OFF to avoid having a soft error inadvertently enabling the module when a single bit flips.

NOTE: Registers that affect the operation of the module, should be only programmed when the BUSY bit is 0 and the ON/OFF bits are not Ah.

NOTE: If the module was in operation, turned off (ON/OFF = all other than Ah) and then turned on (ON/OFF = Ah) again, it is recommended to perform a reset (RESET = 1) of the module before switching it on. This avoids that the state machine is held in an unrecoverable state.

NOTE: A write to these register bits while receiving a packet will not have any effect on the received packet. The mode change will be performed after the packet is received

19.3.2 DMM Interrupt Set Register (DMMINTSET)

This register contains the interrupt set bits for error interrupts and functional interrupts. Only the bits which are relevant for the particular mode (trace mode or direct data mode) will be taken into account for the interrupt generation.

Figure 19-8. DMM Interrupt Set Register (DMMINTSET) [offset = 04h]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved						PROG_BUFF	EO_BUFF
R-0						R/WP-0	R/WP-0

15	14	13	12	11	10	9	8
DEST3REG2	DEST3REG1	DEST2REG2	DEST2REG1	DEST1REG2	DEST1REG1	DEST0REG2	DEST0REG1
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
BUSERROR	BUFF_OVF	SRC_OVF	DEST3_ERR	DEST2_ERR	DEST1_ERR	DEST0_ERR	PACKET_ ERR_INT
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-8. DMM Interrupt Set Register (DMMINTSET) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads returns 0. Writes have no effect.
17	PROG_BUFF		**Programmable Buffer Interrupt Set.** This enables the interrupt generation in case the buffer pointer equals the programmed value in the DMMINTPT register (Section 19.3.11). This bit is only relevant in Direct Data Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on pointer match.
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
16	EO_BUFF		**End of Buffer Interrupt Set.** This enables the interrupt generation in case data was written to the last entry in the buffer and the pointer wrapped around to the beginning of the buffer. This bit is only relevant in Direct Data Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on writing to the last entry.
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
15	DEST3REG2		**Destination 3 Region 2 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 3 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)

Table 19-8. DMM Interrupt Set Register (DMMINTSET) Field Descriptions (continued)

Bit	Field	Value	Description
14	DEST3REG1		**Destination 3 Region 1 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 3 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
13	DEST2REG2		**Destination 2 Region 2 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 2 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
12	DEST2REG1		**Destination 2 Region 1 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 2 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
11	DEST1REG2		**Destination 1 Region 2 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 1 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
10	DEST1REG1		**Destination 1 Region 1 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 1 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
9	DEST0REG2		**Destination 0 Region 2 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 0 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)

Table 19-8. DMM Interrupt Set Register (DMMINTSET) Field Descriptions (continued)

Bit	Field	Value	Description
8	DEST0REG1		**Destination 0 Region 1 Interrupt Set.** This enables the interrupt generation in case data was accessed at the startaddress of Destination 0 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
7	BUSERROR		Bus Error Response for errors generated when doing internal bus transfers.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
6	BUFF_OVF		**Buffer Overflow.** This enables the interrupt generation in case new data is received, while the previous data still has not been transmitted.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
5	SRC_OVF		**Source Overflow.** This enables an interrupt if the external system experienced and overflow which was signalled in the Trace Mode packet.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
4	DEST3_ERR		**Destination 3 Error.** This enables the interrupt generation in case data should be written into a address not specified by DMMDEST3REG1/DMMDEST3BL1 or DMMDEST3REG2/DMMDEST3BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)

Table 19-8. DMM Interrupt Set Register (DMMINTSET) Field Descriptions (continued)

Bit	Field	Value	Description
3	DEST2_ERR		**Destination 2 Error Interrupt Set.** This enables the interrupt generation in case data should be written into a address not specified by DMMDEST2REG1/DMMDEST2BL1 or DMMDEST2REG2/DMMDEST2BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
2	DEST1_ERR		**Destination 1 Error Interrupt Set.** This enables the interrupt generation in case data should be written into a address not specified by DMMDEST1REG1/DMMDEST1BL1 or DMMDEST1REG2/DMMDEST1BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
1	DEST0_ERR		**Destination 0 Error Interrupt Set.** This enables the interrupt generation in case data should be written into a address not specified by DMMDEST0REG1/DMMDEST0BL1 or DMMDEST0REG2/DMMDEST0BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)
0	PACKET_ERR_INT		**Packet Error.** This enables the interrupt generation in case of an error condition in the packet reception. Please refer to Section 19.2.3 for the error conditions.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Enable interrupt (sets corresponding bit in DMMINTCLR; DMMINTLVL)

19.3.3 DMM Interrupt Clear Register (DMMINTCLR)

This register contains the interrupt clear bits for error interrupts and functional interrupts. Only the bits which are relevant for the particular mode (trace mode or direct data mode) will be taken into account for the interrupt generation

Figure 19-9. DMM Interrupt Clear Register (DMMINTCLR) [offset = 08h]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved						PROG_BUFF	EO_BUFF
R-0						R/WP-0	R/WP-0

15	14	13	12	11	10	9	8
DEST3REG2	DEST3REG1	DEST2REG2	DEST2REG1	DEST1REG2	DEST1REG1	DEST0REG2	DEST0REG1
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
BUSERROR	BUFF_OVF	SRC_OVF	DEST3_ERR	DEST2_ERR	DEST1_ERR	DEST0_ERR	PACKET_ ERR_INT
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-9. DMM Interrupt Clear Register (DMMINTCLR) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads returns 0. Writes have no effect.
17	PROG_BUFF		**Programmable Buffer Interrupt Set.**This disables the interrupt generation in case the buffer pointer equals the programmed value in the DMMINTPT register (Section 19.3.11). This bit is only relevant in Direct Data Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on pointer match.
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
16	EO_BUFF		**End of Buffer Interrupt Set.**This disables the interrupt generation in case data was written to the last entry in the buffer and the pointer wrapped around to the beginning of the buffer. This bit is only relevant in Direct Data Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on writing to the last entry.
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))

Table 19-9. DMM Interrupt Clear Register (DMMINTCLR) Field Descriptions (continued)

Bit	Field	Value	Description
15	DEST3REG2		**Destination 3 Region 2 Interrupt Set.**This disables the interrupt generation in case data was accessed at the startaddress of Destination 3 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
14	DEST3REG1		**Destination 3 Region 1 Interrupt Set.**This disables the interrupt generation in case data was accessed at the startaddress of Destination 3 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
13	DEST2REG2		**Destination 2 Region 2 Interrupt Set.**This disables the interrupt generation in case data was accessed at the startaddress of Destination 2 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
12	DEST2REG1		**Destination 2 Region 1 Interrupt Set.**This disables the interrupt generation in case data was accessed at the startaddress of Destination 2 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
11	DEST1REG2		**Destination 1 Region 2 Interrupt Set.**This disables the interrupt generation in case data was accessed at the startaddress of Destination 1 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))

Table 19-9. DMM Interrupt Clear Register (DMMINTCLR) Field Descriptions (continued)

Bit	Field	Value	Description
10	DEST1REG1		**Destination 1 Region 1 Interrupt Set.**This enables the interrupt generation in case data was accessed at the startaddress of Destination 1 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
9	DEST0REG2		**Destination 0 Region 2 Interrupt Set.**This disables the interrupt generation in case data was accessed at the startaddress of Destination 0 Region 2. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
8	DEST0REG1		**Destination 0 Region 1 Interrupt Set.**This disables the interrupt generation in case data was accessed at the startaddress of Destination 0 Region 1. This bit is only relevant in Trace Mode.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated on a write to the start address of this region
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
7	BUSERROR		**Bus Error Response** for errors generated when doing internal bus transfers.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
6	BUFF_OVF		**Buffer Overflow.**This disables the interrupt generation in case new data is received, while the previous data still has not been transmitted.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))

Table 19-9. DMM Interrupt Clear Register (DMMINTCLR) Field Descriptions (continued)

Bit	Field	Value	Description
5	SRC_OVF		**Source Overflow.** This disables an interrupt if the external system experienced and overflow which was signalled in the Trace Mode packet.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
4	DEST3_ERR		**Destination 3 Error.**This disables the interrupt generation in case data should be written into a address not specified by DMMDEST3REG1/DMMDEST3BL1 or DMMDEST3REG2/DMMDEST3BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
3	DEST2_ERR		**Destination 2 Error Interrupt Set.**This disables the interrupt generation in case data should be written into a address not specified by DMMDEST2REG1/DMMDEST2BL1 or DMMDEST2REG2/DMMDEST2BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
2	DEST1_ERR		**Destination 1 Error Interrupt Set.**This disables the interrupt generation in case data should be written into a address not specified by DMMDEST1REG1/DMMDEST1BL1 or DMMDEST1REG2/DMMDEST1BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))

Table 19-9. DMM Interrupt Clear Register (DMMINTCLR) Field Descriptions (continued)

Bit	Field	Value	Description
1	DEST0_ERR		**Destination 0 Error Interrupt Set.**This disables the interrupt generation in case data should be written into a address not specified by DMMDEST0REG1/DMMDEST0BL1 or DMMDEST0REG2/DMMDEST0BL2. If both blocksizes are programmed to 0 or a reserved value, the interrupt will still be generated, the write to the internal RAM however will not take place.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))
0	PACKET_ERR_INT		**Packet Error.**This disables the interrupt generation in case of an error condition in the packet reception. Please refer to Section 19.2.3for the error conditions.
			User and privilege mode (read):
		0	No interrupt will be generated
		1	An interrupt will be generated
			Privilege mode (write):
		0	No influence on bit
		1	Disable interrupt (clears corresponding bit in DMMINTCLR; DMM Interrupt Level Register (DMMINTLVL))

19.3.4 DMM Interrupt Level Register (DMMINTLVL)

This register contains the interrupt level bits for error interrupts and normal interrupts.

Figure 19-10. DMM Interrupt Level Register (DMMINTLVL) [offset = 0Ch]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved						PROG_BUFF	EO_BUFF
R-0						R/WP-0	R/WP-0

15	14	13	12	11	10	9	8
DEST3REG2	DEST3REG1	DEST2REG2	DEST2REG1	DEST1REG2	DEST1REG1	DEST0REG2	DEST0REG1
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
BUSERROR	BUFF_OVF	SRC_OVF	DEST3_ERR	DEST2_ERR	DEST1_ERR	DEST0_ERR	PACKET_ERR_INT
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-10. DMM Interrupt Level Register (DMMINTLVL) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads returns 0. Writes have no effect.
17	PROG_BUFF		**Programmable Buffer Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
16	EO_BUFF		**End of Buffer Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
15	DEST3REG2		**Destination 3 Region 2 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
14	DEST3REG1		**Destination 3 Region 1 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
13	DEST2REG2		**Destination 2 Region 2 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
12	DEST2REG1		**Destination 2 Region 1 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
11	DEST1REG2		**Destination 1 Region 2 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1

Table 19-10. DMM Interrupt Level Register (DMMINTLVL) Field Descriptions (continued)

Bit	Field	Value	Description
10	DEST1REG1		**Destination 1 Region 1 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
9	DEST0REG2		**Destination 0 Region 2 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
8	DEST0REG1		**Destination 0 Region 1 Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
7	BUSERROR		**BMM Bus Error Response**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
6	BUFF_OVF		**Write Buffer Overflow Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
5	SRC_OVF		**Source Overflow Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
4	DEST3_ERR		**Destination 3 Error Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
3	DEST2_ERR		**Destination 2 Error Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
2	DEST1_ERR		**Destination 1 Error Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
1	DEST0_ERR		**Destination 0 Error Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1
0	PACKET_ERR_INT		**Packet Error Interrupt Level**
			User and privilege mode read, privilege mode write:
		0	Interrupt mapped to level 0
		1	Interrupt mapped to level 1

19.3.5 DMM Interrupt Flag Register (DMMINTFLG)

This register contains the interrupt level bits for error interrupts and normal interrupts.

Figure 19-11. DMM Interrupt Flag Register (DMMINTFLG) [offset = 10h]

31							24
Reserved							
R-0							
23						17	16
Reserved						PROG_BUFF	EO_BUFF
R-0						R/WPC-0	R/WPC-0
15	14	13	12	11	10	9	8
DEST3REG2	DEST3REG1	DEST2REG2	DEST2REG1	DEST1REG2	DEST1REG1	DEST0REG2	DEST0REG1
R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0
7	6	5	4	3	2	1	0
BUSERROR	BUFF_OVF	SRC_OVF	DEST3_ERR	DEST2_ERR	DEST1_ERR	DEST0_ERR	PACKET_ERR_INT
R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0	R/WPC-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; C = Clear; -*n* = value after reset

Table 19-11. DMM Interrupt Flag Register (DMMINTFLG) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads returns 0. Writes have no effect.
17	PROG_BUFF		**Programmable Buffer Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
16	EO_BUFF		**End of Buffer Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
15	DEST3REG2		**Destination 3 Region 2 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
14	DEST3REG1		**Destination 3 Region 1 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared

Table 19-11. DMM Interrupt Flag Register (DMMINTFLG) Field Descriptions (continued)

Bit	Field	Value	Description
13	DEST2REG2		**Destination 2 Region 2 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
12	DEST2REG1		**Destination 2 Region 1 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
11	DEST1REG2		**Destination 1 Region 2 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
10	DEST1REG1		**Destination 1 Region 1 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
9	DEST0REG2		**Destination 0 Region 2 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
8	DEST0REG1		**Destination 0 Region 1 Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared

Table 19-11. DMM Interrupt Flag Register (DMMINTFLG) Field Descriptions (continued)

Bit	Field	Value	Description
7	BUSERROR		**BMM Bus Error Response.**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
6	BUFF_OVF		**Write Buffer Overflow Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
5	SRC_OVF		**Source Overflow Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
4	DEST3_ERR		**Destination 3 Error Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
3	DEST2_ERR		**Destination 2 Error Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
2	DEST1_ERR		**Destination 1 Error Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared

Table 19-11. DMM Interrupt Flag Register (DMMINTFLG) Field Descriptions (continued)

Bit	Field	Value	Description
1	DEST0_ERR		**Destination 0 Error Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared
0	PACKET_ERR_INT		**Packet Error Interrupt Flag**
			User and privilege mode (read):
		0	No interrupt occurred
		1	Interrupt occurred
			Privilege mode (write):
		0	No influence on bit
		1	Bit will be cleared

19.3.6 DMM Interrupt Offset 1 Register (DMMOFF1)

This register holds the offset indicating which interrupt occurred on interrupt level 0. The CPU can read this register to determine the source of the interrupt without having to test individual interrupt flags.

Figure 19-12. DMM Interrupt Offset 1 Register (DMMOFF1) [offset = 14h]

31	16
Reserved	
R-0	

15	5	4	0
Reserved		OFFSET	
R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 19-12. DMM Interrupt Offset 1 Register (DMMOFF1) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Read returns 0. Writes have no effect.
4-0	OFFSET		User and privilege mode (read):
		Bit Encoding	Interrupt
		0	Phantom. All interrupt flags have been cleared before the offset register has been read.
		1h	Packet Error
		2h	Destination 0 Error
		3h	Destination 1 Error
		4h	Destination 2 Error
		5h	Destination 3 Error
		6h	Source Overflow
		7h	Buffer Overflow
		8h	Bus Error
		9h	Destination 0 Region 1
		Ah	Destination 0 Region 2
		Bh	Destination 1 Region 1
		Ch	Destination 1 Region 2
		Dh	Destination 2 Region 1
		Eh	Destination 2 Region 2
		Fh	Destination 3 Region 1
		10h	Destination 3 Region 2
		11h	End of Buffer
		12h	Programmable Buffer
		13h-1Fh	Reserved
			Reading the offset will clear the corresponding flag in DMMINTFLG (Section 19.3.5).
			Privilege and user mode writes have no effect

19.3.7 DMM Interrupt Offset 2 Register (DMMOFF2)

This register holds the offset indicating which interrupt occurred on interrupt level 1. The CPU can read this register to determine the source of the interrupt without having to test individual interrupt flags.

Figure 19-13. DMM Interrupt Offset 2 Register (DMMOFF2) [offset = 18h]

31	16
Reserved	
R-0	

15	5	4	0
Reserved		OFFSET	
R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 19-13. DMM Interrupt Offset 2 Register (DMMOFF1) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Read returns 0. Writes have no effect.
4-0	OFFSET		User and privilege mode (read):
		Bit Encoding	Interrupt
		0	Phantom. All interrupt flags have been cleared before the offset register has been read.
		1h	Packet Error
		2h	Destination 0 Error
		3h	Destination 1 Error
		4h	Destination 2 Error
		5h	Destination 3 Error
		6h	Source Overflow
		7h	Buffer Overflow
		8h	Bus Error
		9h	Destination 0 Region 1
		Ah	Destination 0 Region 2
		Bh	Destination 1 Region 1
		Ch	Destination 1 Region 2
		Dh	Destination 2 Region 1
		Eh	Destination 2 Region 2
		Fh	Destination 3 Region 1
		10h	Destination 3 Region 2
		11h	End of Buffer
		12h	Programmable Buffer
		13h-1Fh	Reserved
			Reading the offset will clear the corresponding flag in DMMINTFLG (Section 19.3.5).
			Privilege and user mode writes have no effect

19.3.8 DMM Direct Data Mode Destination Register (DMMDDMDEST)

This register defines the starting address of the buffer used to store the received data in Direct Data Mode. By writing to this register, the DMMDDMPT register (Section 19.3.10) will be set to 0x0000.

Figure 19-14. DMM Direct Data Mode Destination Register (DMMDDMDEST) [offset = 1Ch]

31	0
STARTADDR	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-14. DMM Direct Data Mode Destination Register (DMMDDMDEST) Field Descriptions

Bit	Field	Description
31-0	STARTADDR	These bits define the starting address of the buffer. The starting address has to be a multiple of the blocksize chosen in DMMDDMBL (Section 19.3.9). User and privilege mode (read): current start address Privilege mode (write): sets start address to value written

19.3.9 DMM Direct Data Mode Blocksize Register (DMMDDMBL)

This register defines the blocksize of the buffer used to store the received data in Direct Data Mode.

Figure 19-15. DMM Direct Data Mode Blocksize Register (DMMDDMBL) [offset = 20h]

31	16
Reserved	
R-0	

15	4	3	0
Reserved		BLOCKSIZE	
R-0		R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-15. DMM Direct Data Mode Blocksize Register (DMMDDMBL) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Read returns 0. Writes have no effect.
3-0	BLOCKSIZE		These bits define the size of the buffer region User and privilege mode (read): current block size Privilege mode (write):
		0	Buffer disabled. No data will be stored.
		1h	32 Byte
		2h	64 Byte
		3h	128 Byte
		4h	256 Byte
		5h	512 Byte
		6h	1 KByte
		7h	2 KByte
		8h	4 KByte
		9h	8 KByte
		Ah	16 KByte
		Bh	32 KByte
		Ch-Fh	Reserved

19.3.10 DMM Direct Data Mode Pointer Register (DMMDDMPT)

This register shows the pointer into the buffer programmed by DMMDDMDEST (Section 19.3.8) and DMMDDMBL (Section 19.3.9).

Figure 19-16. DMM Direct Data Mode Pointer Register (DMMDDMPT) [offset = 24h]

31	16
Reserved	
R-0	

15	14	0
Rsvd	POINTER	
R-0	R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 19-16. DMM Direct Data Mode Pointer Register (DMMDDMPT) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Read returns 0. Writes have no effect.
14-0	POINTER		These bits hold the pointer to the next entry to be written in the buffer. The pointer points to the byte aligned address. If in 16-bit DDM mode, bit 0 will be 0. If in 32-bit DDM mode, bit 0 and 1 will be 0. User and privilege mode (read): next data entry Privilege mode (write): writes have no effect

19.3.11 DMM Direct Data Mode Interrupt Pointer Register (DMMINTPT)

This register can be programmed to hold a threshold to which the DMMDDMPT register (Section 19.3.10) is compared. An interrupt can be generated when both match.

Figure 19-17. DMM Direct Data Mode Interrupt Pointer Register (DMMINTPT) [offset = 28h]

31	16
Reserved	
R-0	

15	14	0
Rsvd	INTPT	
R-0	R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-17. DMM Direct Data Mode Interrupt Pointer Register (DMMINTPT) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Read returns 0. Writes have no effect.
14-0	INTPT		Interrupt Pointer. When the buffer pointer (Section 19.3.10) matches the programmed value in DMMINTPT and the PROG_BUF interrupt (Section 19.3.2) is set, an interrupt is generated. User and privilege mode (read): current interrupt threshold Privilege mode (write): new interrupt threshold

19.3.12 DMM Destination x Region 1 (DMMDESTxREG1)

This register defines the starting address of the buffer used to store the received data in Trace Mode. If the received data does not fall into the address range defined by DMMDESTxREG1 and DMMDESTxBL1, an interrupt (DESTx_ERR) can be generated. The description below is valid for following registers: DMMDEST0REG1, DMMDEST1REG1, DMMDEST2REG1, DMMDEST3REG1.

Figure 19-18. DMM Destination x Region 1 (DMMDESTxREG1) [offset = 2Ch, 3Ch, 4Ch, 5Ch]

31	18	17	16
BASEADDR		BLOCKADDR	
R/WP-0		R/WP-0	

15	0
BLOCKADDR	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-18. DMM Destination x Region 1 (DMMDESTxREG1) Field Descriptions

Bit	Field	Description
31-18	BASEADDR	These bits define the base address of the 256kB region where the buffer is located. User and privilege mode (read): current start address Privilege mode (write): sets base address to value written
17-0	BLOCKADDR	These bits define the starting address of the buffer in the 256kB page. The starting address has to be a multiple of the blocksize chosen in DMMDESTxBL1 (Section 19.3.13). User and privilege mode (read): current start address Privilege mode (write): sets start address to value written

19.3.13 DMM Destination x Blocksize 1 (DMMDESTxBL1)

This register defines the blocksize of the buffer used to store the received data in Trace Mode. If the received data does not fall into the address range defined by DMMDESTxREG1 and DMMDESTxBL1, an interrupt (DESTx_ERR) can be generated. The description below is valid for following registers: DMMDEST0BL1, DMMDEST1BL1, DMMDEST2BL1, DMMDEST3BL1.

Figure 19-19. DMM Destination x Blocksize 1 (DMMDESTxBL1) [offset = 30h, 40h, 50h, 60h]

31	16
Reserved	
R-0	

15	4	3	0
Reserved		BLOCKSIZE	
R-0		R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-19. DMM Destination x Blocksize 1 (DMMDESTxBL1) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Read returns 0. Writes have no effect.
3-0	BLOCKSIZE		These bits define the length of the buffer region. If all bits are 0, the region is disabled and no data will be stored.
			User and privilege mode (read): current block size
			Privilege mode (write):
		0	Region disabled
		1h	1 KByte
		2h	2 KByte
		3h	4 KByte
		4h	8 KByte
		5h	16 KByte
		6h	32 KByte
		7h	64 KByte
		8h	128 KByte
		9h	256 KByte
		Ah-Fh	Reserved

19.3.14 DMM Destination x Region 2 (DMMDESTxREG2)

This register defines the starting address of the buffer used to store the received data in Trace Mode. If the received data does not fall into the address range defined by DMMDESTxREG2 and DMMDESTxBL2, an interrupt (DESTx_ERR) can be generated. The description below is valid for following registers: DMMDEST0REG2, DMMDEST1REG2, DMMDEST2REG2, DMMDEST3REG2.

Figure 19-20. DMM Destination x Region 2 (DMMDESTxREG2) [offset = 34h, 44h, 54h, 64h]

31	18	17	16
BASEADDR		BLOCKADDR	
R/WP-0		R/WP-0	

15	0
BLOCKADDR	
R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-20. DMM Destination x Region 2 (DMMDESTxREG2) Field Descriptions

Bit	Field	Description
31-18	BASEADDR	These bits define the base address of the 256kB region where the buffer is located. User and privilege mode (read): current start address Privilege mode (write): sets base address to value written
17-0	BLOCKADDR	These bits define the starting address of the buffer in the 256kB page. The starting address has to be a multiple of the blocksize chosen in DMMDESTxBL1 (Section 19.3.15). User and privilege mode (read): current start address Privilege mode (write): sets start address to value written

19.3.15 DMM Destination x Blocksize 2 (DMMDESTxBL2)

This register defines the blocksize of the buffer used to store the received data in Trace Mode. If the received data does not fall into the address range defined by DMMDESTxREG2 and DMMDESTxBL2, an interrupt (DESTx_ERR) can be generated. The description below is valid for following registers: DMMDEST0BL2, DMMDEST1BL2, DMMDEST2BL2, DMMDEST3BL2.

Figure 19-21. DMM Destination x Blocksize 2 (DMMDESTxBL2) [offset = 38h, 48h, 58h, 68h]

31	16
Reserved	
R-0	

15	4	3	0
Reserved		BLOCKSIZE	
R-0		R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 19-21. DMM Destination x Blocksize 2 (DMMDESTxBL2) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Read returns 0. Writes have no effect.
3-0	BLOCKSIZE		These bits define the length of the buffer region. If all bits are 0, the region is disabled and no data will be stored.
			User and privilege mode (read): current block size
			Privilege mode (write):
		0	Region disabled
		1h	1 KByte
		2h	2 KByte
		3h	4 KByte
		4h	8 KByte
		5h	16 KByte
		6h	32 KByte
		7h	64 KByte
		8h	128 KByte
		9h	256 KByte
		Ah-Fh	Reserved

19.3.16 DMM Pin Control 0 (DMMPC0)

This register defines if the DMM pins are used in functional or GIO mode. It should only be written when ON/OFF = 0101 and the BUSY bit = 0 (Section 19.3.1). If pins other than the pins specified in Table 19-5 are configured, or DMMCLK and DMMSYNC are programmed as non-functional pins, no operation in trace mode or direct data mode is possible.

Figure 19-22. DMM Pin Control 0 (DMMPC0) [offset = 6Ch]

31	24
Reserved	
R-0	

23	19	18	17	16
Reserved		ENAFUNC	DATA15FUNC	DATA14FUNC
R-0		R/WP-0	R/WP-0	R/WP-0

15	14	13	12	11	10	9	8
DATA13FUNC	DATA12FUNC	DATA11FUNC	DATA10FUNC	DATA9FUNC	DATA8FUNC	DATA7FUNC	DATA6FUNC
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
DATA5FUNC	DATA4FUNC	DATA3FUNC	DATA2FUNC	DATA1FUNC	DATA0FUNC	CLKFUNC	SYNCFUNC
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-22. DMM Pin Control 0 (DMMPC0) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENAFUNC		**Functional mode of $\overline{\text{DMMENA}}$ pin.** This bit defines whether the pin is used in functional mode or in GIO mode.
			User and privilege mode (read):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode
			Privilege mode (write):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode
17-2	DATAxFUNC		**Functional mode of DMMDATA[x] pin.** This bit defines whether the pin is used in functional mode or in GIO mode. If pins are configured in functional mode, only pins defined in Table 19-5 have to be used for proper operation.
			User and privilege mode (read):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode
			Privilege mode (write):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode
1	CLKFUNC		**Functional mode of DMMCLK pin.** This bit defines whether the pin is used in functional mode or in GIO mode.
			User and privilege mode (read):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode
			Privilege mode (write):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode

Table 19-22. DMM Pin Control 0 (DMMPC0) Field Descriptions (continued)

Bit	Field	Value	Description
0	SYNCFUNC		**Functional mode of DMMSYNC pin.** This bit defines whether the pin is used in functional mode or in GIO mode.
			User and privilege mode (read):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode
			Privilege mode (write):
		0	Pin is used in GIO mode
		1	Pin is used in Functional mode

19.3.17 DMM Pin Control 1 (DMMPC1)

The bits in this register define the direction of the individual module pins when in GIO mode.

Figure 19-23. DMM Pin Control 1 (DMMPC1) [offset = 70h]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved					ENADIR	DATA15DIR	DATA14DIR
R-0					R/WP-0	R/WP-0	R/WP-0

15	14	13	12	11	10	9	8
DATA13DIR	DATA12DIR	DATA11DIR	DATA10DIR	DATA9DIR	DATA8DIR	DATA7DIR	DATA6DIR
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
DATA5DIR	DATA4DIR	DATA3DIR	DATA2DIR	DATA1DIR	DATA0DIR	CLKDIR	SYNCDIR
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-23. DMM Pin Control 1 (DMMPC1) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENADIR		**Direction of $\overline{\text{DMMENA}}$ pin**.
			User and privilege mode (read):
		0	Pin is used as input
		1	Pin is used as output
			Privilege mode (write):
		0	Pin is set to input
		1	Pin is set to output
17-2	DATAxDIR		**Direction of DMMDATA[x] pin.** This bit defines whether the pin is used as input or output in GIO mode.
			User and privilege mode (read):
		0	Pin is used as input
		1	Pin is used as output
			Privilege mode (write):
		0	Pin is set to input
		1	Pin is set to output

Table 19-23. DMM Pin Control 1 (DMMPC1) Field Descriptions (continued)

Bit	Field	Value	Description
1	CLKDIR		**Direction of DMMCLK pin.** This bit defines whether the pin is used as input or output in GIO mode.
			User and privilege mode (read):
		0	Pin is used as input
		1	Pin is used as output
			Privilege mode (write):
		0	Pin is set to input
		1	Pin is set to output
0	SYNCDIR		**Direction of DMMSYNC pin.** This bit defines whether the pin is used as input or output in GIO mode.
			User and privilege mode (read):
		0	Pin is used as input
		1	Pin is used as output
			Privilege mode (write):
		0	Pin is set to input
		1	Pin is set to output

19.3.18 DMM Pin Control 2 (DMMPC2)

The bits in this register reflect the digital representation of the voltage level at the module pins. Even if a pin is configured to be an output pin, the level can be read back via this register.

Figure 19-24. DMM Pin Control 2 (DMMPC2) [offset = 74h]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved					ENAIN	DATA15IN	DATA14IN
R-0					R/WP-0	R/WP-0	R/WP-0

15	14	13	12	11	10	9	8
DATA13IN	DATA12IN	DATA11IN	DATA10IN	DATA9IN	DATA8IN	DATA7IN	DATA6IN
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
DATA5IN	DATA4IN	DATA3IN	DATA2IN	DATA1IN	DATA0IN	CLKIN	SYNCIN
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-24. DMM Pin Control 2 (DMMPC2) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENAIN		**$\overline{\text{DMMENA}}$ input.** This bit reflects the state of the pin in all modes.
			User and privilege mode (read):
		0	Logic low (input voltage is V_{IL} or lower)
		1	Logic high (input voltage is V_{IH} or higher)
			Privilege mode (write): writes to this bit have no effect.
17-2	DATAxIN		**DMMDATA[x] input.** This bit reflects the state of the pin in all modes.
			User and privilege mode (read):
		0	Logic low (input voltage is V_{IL} or lower)
		1	Logic high (input voltage is V_{IH} or higher)
			Privilege mode (write): writes to this bit have no effect.
1	CLKIN		**DMMCLK input.** This bit reflects the state of the pin in all modes.
			User and privilege mode (read):
		0	Logic low (input voltage is V_{IL} or lower)
		1	Logic high (input voltage is V_{IH} or higher)
			Privilege mode (write): writes to this bit have no effect.
0	SYNCIN		**DMMSYNC input.** This bit reflects the state of the pin in all modes
			User and privilege mode (read):
		0	Logic low (input voltage is V_{IL} or lower)
		1	Logic high (input voltage is V_{IH} or higher)
			Privilege mode (write): writes to this bit have no effect.

19.3.19 DMM Pin Control 3 (DMMPC3)

The bits in this register set the pin to logic low or high level if the pin is configured as output (Section 19.3.17).

Figure 19-25. DMM Pin Control 3 (DMMPC3) [offset = 78h]

31	30	29	28	27	26	25	24
Reserved							
R-0							
23	22	21	20	19	18	17	16
Reserved					ENAOUT	DATA15OUT	DATA14OUT
R-0					R/WP-0	R/WP-0	R/WP-0
15	14	13	12	11	10	9	8
DATA13OUT	DATA12OUT	DATA11OUT	DATA10OUT	DATA9OUT	DATA8OUT	DATA7OUT	DATA6OUT
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0
7	6	5	4	3	2	1	0
DATA5OUT	DATA4OUT	DATA3OUT	DATA2OUT	DATA1OUT	DATA0OUT	CLKOUT	SYNCOUT
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-25. DMM Pin Control 3 (DMMPC3) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENAOUT		**Output state of $\overline{\text{DMMENA}}$ pin.** This bit sets the pin to logic low or high level
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	Logic low (output voltage is set to V_{OL} or lower)
		1	Logic high (output voltage is set to V_{OH} or higher)
17-2	DATAxOUT		**Output state of DMMDATA[x] pin.** This bit sets the pin to logic low or high level.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	Logic low (output voltage is set to V_{OL} or lower)
		1	Logic high (output voltage is set to V_{OH} or higher)
1	CLKOUT		**Output state of DMMCLK pin.** This bit sets the pin to logic low or high level
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	Logic low (output voltage is set to V_{OL} or lower)
		1	Logic high (output voltage is set to V_{OH} or higher)

Table 19-25. DMM Pin Control 3 (DMMPC3) Field Descriptions (continued)

Bit	Field	Value	Description
0	SYNCOUT		**Output state of DMMSYNC pin.** This bit sets the pin to logic low or high level.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	Logic low (output voltage is set to V_{OL} or lower)
		1	Logic high (output voltage is set to V_{OH} or higher)

19.3.20 DMM Pin Control 4 (DMMPC4)

This register allows to set individual pins to a logic high level without having to do a read-modify-write operation as would be the case with the DMMPC3 register (Section 19.3.19). Writing a zero to a bit will not change the state of the pin.

Figure 19-26. DMM Pin Control 4 (DMMPC4) [offset = 7Ch]

31							24
Reserved							
R-0							
23					19	18	17	16
Reserved						ENASET	DATA15SET	DATA14SET
R-0						R/WP-0	R/WP-0	R/WP-0
15	14	13	12	11	10	9	8
DATA13SET	DATA12SET	DATA11SET	DATA10SET	DATA9SET	DATA8SET	DATA7SET	DATA6SET
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0
7	6	5	4	3	2	1	0
DATA5SET	DATA4SET	DATA3SET	DATA2SET	DATA1SET	DATA0SET	CLKSET	SYNCSET
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-26. DMM Pin Control 4 (DMMPC4) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENASET		**Sets output state of $\overline{\text{DMMENA}}$ pin to logic high.** Value in the ENASET bit sets the data output control register bit to 1 regardless of the current value in the ENAOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Logic high (output voltage is set to V_{OH} or higher)
17-2	DATAxSET		**Sets output state of DMMDATA[x] pin to logic high.** Value in the DATAxSET bit sets the data output control register bit to 1 regardless of the current value in the DATAxOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Logic high (output voltage is set to V_{OH} or higher)

Table 19-26. DMM Pin Control 4 (DMMPC4) Field Descriptions (continued)

Bit	Field	Value	Description
1	CLKSET		**Sets output state of DMMCLK pin to logic high.** Value in the CLKSET bit sets the data output control register bit to 1 regardless of the current value in the CLKOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Logic high (output voltage is set to V_{OH} or higher)
0	SYNCSET		**Sets output state of DMMSYNC pin logic high.** Value in the SYNCSET bit sets the data output control register bit to 1 regardless of the current value in the SYNCOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Logic high (output voltage is set to V_{OH} or higher)

19.3.21 DMM Pin Control 5 (DMMPC5)

This register allows to set individual pins to a logic low level without having to do a read-modify-write operation as would be the case with the DMMPC3 register (Section 19.3.19). Writing a one to a bit will change the output to a logic low level, writing a zero will not change the state of the pin.

Figure 19-27. DMM Pin Control 5 (DMMPC5) [offset = 80h]

31							24
Reserved							
R-0							
23					19	18	17	16
Reserved						ENACLR	DATA15CLR	DATA14CLR
R-0						R/WP-0	R/WP-0	R/WP-0
15	14	13	12	11	10	9	8
DATA13CLR	DATA12CLR	DATA11CLR	DATA10CLR	DATA9CLR	DATA8CLR	DATA7CLR	DATA6CLR
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0
7	6	5	4	3	2	1	0
DATA5CLR	DATA4CLR	DATA3CLR	DATA2CLR	DATA1CLR	DATA0CLR	CLKCLR	SYNCCLR
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-27. DMM Pin Control 5 (DMMPC5) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENACLR		**Sets output state of $\overline{\text{DMMENA}}$ pin to logic low.** Value in the ENACLR bit clears the data output control register bit to 0, regardless of the current value in the ENAOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Clears the pin to logic low (output voltage is set to V_{OL} or lower)
17-2	DATAxCLR		**Sets output state of DMMDATA[x] pin to logic low.** Value in the DATAxCLR bit clears the data output control register DATAxOUT bit to 0, regardless of the current value in the DATAxOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Clears the pin to logic low (output voltage is set to V_{OL} or lower)
1	CLKCLR		**Sets output state of DMMCLK pin to logic low.** Value in the CLKCLR bit clears the data output control register CLKOUT bit to 0, regardless of the current value in the CLKOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Clears the pin to logic low (output voltage is set to V_{OL} or lower)

Table 19-27. DMM Pin Control 5 (DMMPC5) Field Descriptions (continued)

Bit	Field	Value	Description
0	SYNCCLR		**Sets output state of DMMSYNC pin to logic low.** Value in the SYNCCLR bit clears the data output control register SYNCOUT bit to 0, regardless of the current value in the SYNCOUT bit.
			User and privilege mode (read):
		0	Logic low (output voltage is V_{OL} or lower)
		1	Logic high (output voltage is V_{OH} or higher)
			Privilege mode (write):
		0	State of the pin is unchanged
		1	Clears the pin to logic low (output voltage is set to V_{OL} or lower)

19.3.22 DMM Pin Control 6 (DMMPC6)

These bits configure the pins in push-pull or open-drain functionality. If configured to be open-drain, the module only drives a logic-low level on the pin. An external pull-up resistor needs to be connected to the pin to pull it high, when the pin is in high-impedance mode.

Figure 19-28. DMM Pin Control 6 (DMMPC6) [offset = 84h]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved					ENAPDR	DATA15PDR	DATA14PDR
R-0					R/WP-0	R/WP-0	R/WP-0

15	14	13	12	11	10	9	8
DATA13PDR	DATA12PDR	DATA11PDR	DATA10PDR	DATA9PDR	DATA8PDR	DATA7PDR	DATA6PDR
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	6	5	4	3	2	1	0
DATA5PDR	DATA4PDR	DATA3PDR	DATA2PDR	DATA1PDR	DATA0PDR	CLKPDR	SYNCPDR
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-28. DMM Pin Control 6 (DMMPC6) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENAPDR		**Open Drain enable.** Enables open-drain functionality, if the pin is configured as GIO output (DMMPC0[18] = 0; DMMPC1[18] = 1). If the pin is configured as a functional pin (DMMPC0[18] = 1), the open-drain functionality is disabled.
			User and privilege mode (read):
		0	Pin behaves as normal push/pull pin
		1	Pin operates in open-drain mode
			Privilege mode (write):
		0	Configures pin as push/pull
		1	Configures pin as open drain

Table 19-28. DMM Pin Control 6 (DMMPC6) Field Descriptions (continued)

Bit	Field	Value	Description
17-2	DATAxPDR		**Open Drain enable.** Enables open-drain functionality on pin, if pin is configured as GIO output (DMMPC0[x] = 0; DMMPC1[x] = 1). If the pin is configured as a functional pin (DMMPC0[x] = 1), the open-drain functionality is disabled.
			User and privilege mode (read):
		0	Pin behaves as normal push/pull pin
		1	Pin operates in open-drain mode
			Privilege mode (write):
		0	Configures the pin as push/pull
		1	Configures the pin as open drain
1	CLKPDR		**Open Drain enable.** Enables open-drain functionality on pin, if pin is configured as GIO output (DMMPC0[1] = 0; DMMPC1[1] = 1). If the pin is configured as a functional pin (DMMPC0[1] = 1), the open-drain functionality is disabled.
			User and privilege mode (read):
		0	Pin behaves as normal push/pull pin
		1	Pin operates in open-drain mode
			Privilege mode (write):
		0	Configures the pin as push/pull
		1	Configures the pin as open drain
0	SYNCPDR		**Open Drain enable.** Enables open-drain functionality on pin, if pin is configured as GIO output (DMMPC0[0] = 0; DMMPC1[0] = 1). If the pin is configured as a functional pin (DMMPC0[0] = 1), the open-drain functionality is disabled.
			User and privilege mode (read):
		0	Pin behaves as normal push/pull pin
		1	Pin operates in open-drain mode
			Privilege mode (write):
		0	Configures the pin as push/pull
		1	Configures the pin as open drain

19.3.23 DMM Pin Control 7 (DMMPC7)

The bits in register control the pullup/down functionality of a pin. The internal pullup/down can be enabled or disabled by this register. The reset configuration of these bits is device implementation dependent. Please consult the device datasheet for this information.

Figure 19-29. DMM Pin Control 7 (DMMPC7) [offset = 88h]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved					ENAPDIS	DATA15PDIS	DATA14PDIS
R-0					R/WP-x	R/WP-x	R/WP-x

15	14	13	12	11	10	9	8
DATA13PDIS	DATA12PDIS	DATA11PDIS	DATA10PDIS	DATA9PDIS	DATA8PDIS	DATA7PDIS	DATA6PDIS
R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x

7	6	5	4	3	2	1	0
DATA5PDIS	DATA4PDIS	DATA3PDIS	DATA2PDIS	DATA1PDIS	DATA0PDIS	CLKPDIS	SYNCPDIS
R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x	R/WP-x

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-29. DMM Pin Control 7 (DMMPC7) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENAPDIS		**Pull disable.** Removes internal pullup/pulldown functionality from pin when configured as input pin (DMMPC1[18] = 0).
			User and privilege mode (read):
		0	Pullup/pulldown functionality enabled
		1	Pullup/pulldown functionality disabled
			Privilege mode (write):
		0	Enables pullup/pulldown functionality
		1	Disables pullup/pulldown functionality
17-2	DATAxPDIS		**Pull disable.** Removes internal pullup/pulldown functionality from pin when configured as input pin (DMMPC1[x] = 0).
			User and privilege mode (read):
		0	Pullup/pulldown functionality enabled
		1	Pullup/pulldown functionality disabled
			Privilege mode (write):
		0	Enables pullup/pulldown functionality
		1	Disables pullup/pulldown functionality
1	CLKPDIS		**Pull disable.** Removes internal pullup/pulldown functionality from pin when configured as input pin (DMMPC1[1] = 0).
			User and privilege mode (read):
		0	Pullup/pulldown functionality enabled
		1	Pullup/pulldown functionality disabled
			Privilege mode (write):
		0	Enables pullup/pulldown functionality
		1	Disables pullup/pulldown functionality

Table 19-29. DMM Pin Control 7 (DMMPC7) Field Descriptions (continued)

Bit	Field	Value	Description
0	SYNCPDIS		**Pull disable.** Removes internal pullup/pulldown functionality from pin when configured as input pin (DMMPC1[0] = 0).
			User and privilege mode (read):
		0	Pullup/pulldown functionality enabled
		1	Pullup/pulldown functionality disabled
			Privilege mode (write):
		0	Enables pullup/pulldown functionality
		1	Disables pullup/pulldown functionality

19.3.24 DMM Pin Control 8 (DMMPC8)

These bits control if the internal pullup or pulldown is configured on the input pin.

Figure 19-30. DMM Pin Control 8 (DMMPC8) [offset = 8Ch]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved					ENAPSEL	DATA15PSEL	DATA14PSEL
R-0					R/WP-1	R/WP-1	R/WP-1

15	14	13	12	11	10	9	8
DATA13PSEL	DATA12PSEL	DATA11PSEL	DATA10PSEL	DATA9PSEL	DATA8PSEL	DATA7PSEL	DATA6PSEL
R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1

7	6	5	4	3	2	1	0
DATA5PSEL	DATA4PSEL	DATA3PSEL	DATA2PSEL	DATA1PSEL	DATA0PSEL	CLKPSEL	SYNCPSEL
R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1	R/WP-1

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 19-30. DMM Pin Control 8 (DMMPC8) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads returns 0. Writes have no effect.
18	ENAPSEL		**Pull select.** Configures pullup or pulldown functionality if DMMPC7[18] = 0.
			User and privilege mode (read):
		0	Pulldown functionality enabled
		1	Pullup functionality enabled
			Privilege mode (write):
		0	Enables pulldown functionality
		1	Enables pullup functionality
17-2	DATAxPSEL		**Pull select.** Configures pullup or pulldown functionality if DMMPC7[x] = 0.
			User and privilege mode (read):
		0	Pulldown functionality enabled
		1	Pullup functionality enabled
			Privilege mode (write):
		0	Enables pulldown functionality
		1	Enables pullup functionality

Table 19-30. DMM Pin Control 8 (DMMPC8) Field Descriptions (continued)

Bit	Field	Value	Description
1	CLKPSEL		**Pull select.** Configures pullup or pulldown functionality if DMMPC7[1] = 0.
			User and privilege mode (read):
		0	Pulldown functionality enabled
		1	Pullup functionality enabled
			Privilege mode (write):
		0	Enables pulldown functionality
		1	Enables pullup functionality
0	SYNCPSEL		**Pull select.** Configures pullup or pulldown functionality if DMMPC7[0] = 0.
			User and privilege mode (read):
		0	Pulldown functionality enabled
		1	Pullup functionality enabled
			Privilege mode (write):
		0	Enables pulldown functionality
		1	Enables pullup functionality

Enhanced Pulse Width Modulator (ePWM)

The enhanced pulse width modulator (ePWM) peripheral is a key element in controlling many of the power electronic systems found in both commercial and industrial equipments. The features supported by the ePWM make it especially suitable for digital motor control.

Topic **Page**

20.1 Introduction **2307**
20.2 ePWM Submodules **2311**
20.3 Application Examples **2366**
20.4 MSS_ETPWM1 Registers **2381**

20.1 Introduction

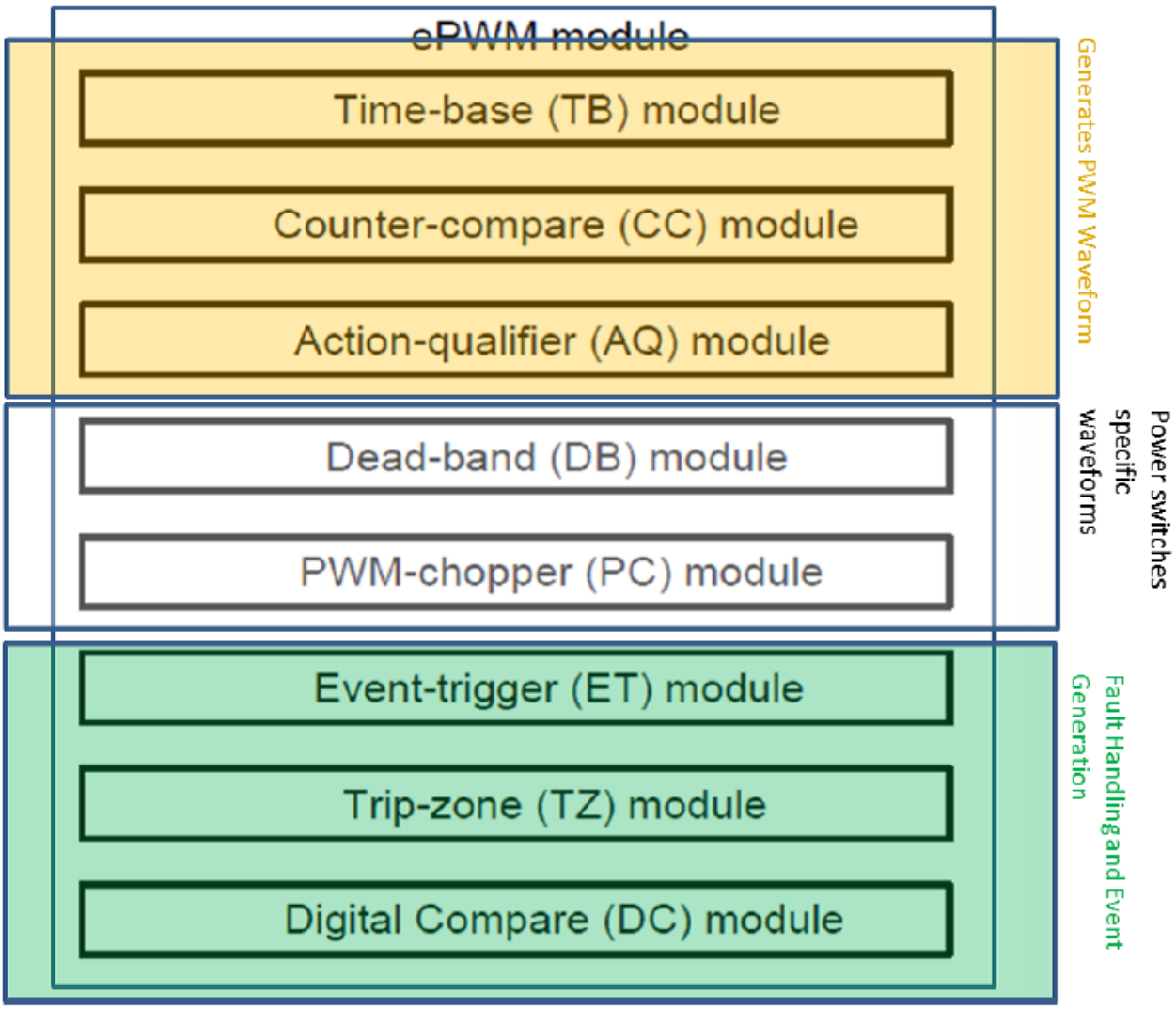
An effective PWM peripheral must be able to generate complex pulse-width waveforms with minimal CPU overhead or intervention. It is highly programmable, flexible, and easy to understand and use. The ePWM is built up from smaller single-channel modules, with separate resources that can operate together as required to form a system. This modular approach results in an orthogonal architecture, and provides a more transparent view of the peripheral structure, helping users to understand its operation quickly.

20.1.1 Submodule Overview

In this section, the letter x within a signal or module name is used to indicate a generic ePWM instance on a device. For example, output signals EPWMxA and EPWMxB refer to the output signals from the ePWMx Instance. Thus, EPWM1A and EPWM1B belong to ePWM1, and likewise EPWM3A and EPWM3B belong to PWM3.

Each ePWM module consists of eight submodules and is connected within a system via the signals shown in Figure 20-1.

Figure 20-1. ePWM Module



The ePWM module represents one complete PWM channel composed of two PWM outputs: EPWMxA and EPWMxB. Multiple ePWM modules are instanced within a device as shown in . Each ePWM instance is identical and is indicated by a numerical value starting with 1. For example, ePWM1 is the first instance and ePWM3 is the third instance in the system, and ePWMx indicates any instance.

The ePWM modules are chained together via a clock synchronization scheme that allows them to operate as a single system when required.

Each ePWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Two PWM outputs (EPWMxA and EPWMxB) that can be used in the following configurations:
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other ePWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- All events can trigger both CPU interrupts and ADC start of conversion (SOC)
- Programmable event prescaling minimizes CPU overhead on interrupts.
- PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.

The main signals used by the ePWM module are:

- **PWM output signals (EPWMxA and EPWMxB).**

  The PWM output signals are made available external to the device through the I/O Multiplexing Module (IOMM) as described in the IOMM chapter.
- **Trip-zone signals ($\overline{TZ1}$ to $\overline{TZ6}$).**

  These input signals alert the ePWM module of fault conditions external to the ePWM module. Each ePWM module can be configured to either use or ignore any of the trip-zone signals. The $\overline{TZ1}$ to $\overline{TZ3}$ trip-zone signals can be configured as asynchronous inputs, or double-synchronized using VCLK3, or double-synchronized and filtered through a 6-VCLK3-cycle counter before connecting to the ePWM modules. This selection is done by configuring registers in the IOMM. $\overline{TZ4}$ is connected to an inverted eQEP1 error signal (EQEP1ERR), or to an inverted eQEP2 error signal (EQEP2ERR), or an OR-combination of EQEP1ERR and EQEP2ERR. This selection is also done via the IOMM registers. $\overline{TZ5}$ is connected to the system clock fail status. This is asserted whenever an oscillator failure is detected, or a PLL slip is detected. $\overline{TZ6}$ is connected to the debug mode entry indicator output from the CPU. This allows you to configure a trip action when the CPU halts.
- **Time-base synchronization input (EPWMxSYNCI) and output (EPWMxSYNCO) signals**.

  The synchronization signals daisy chain the ePWM modules together. Each module can be configured to either use or ignore its synchronization input. The clock synchronization input and output signal are brought out to pins only for ePWM1 (ePWM module #1). The synchronization output for ePWM1 (EPWM1SYNCO) is also connected to the SYNCI of the first enhanced capture module (eCAP1).
- **ADC start-of-conversion signals (EPWMxSOCA and EPWMxSOCB).**

  Each ePWM module has two ADC start of conversion signals. Any ePWM module can trigger a start of conversion. Which event triggers the start of conversion is configured in the Event-Trigger submodule of the ePWM.
- **Peripheral Bus**

  The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.

Each ePWM module is connected to the input/output signals shown in .

NOTE: The integration of the three ePWM modules is done such that it could be used for complex asymmetric waveforms generation for complicated power management systems, however, currently the plan is to use the three modules in a completely independent way, generating a simple symmetric waveform signal for both the PMIC device and the monitoring system input.

20.1.2 Register Mapping

The complete ePWM module control and status register set is grouped by submodule as shown in Table 20-1. Each register set is duplicated for each instance of the ePWM module.

Table 20-1. ePWM Module Control and Status Register Set Grouped by Submodule

Name	Address Offset[1]	Size (x16)	Shadow	Privileged Mode Write Only?	Description
Time-Base Submodule Registers					
TBSTS		1	No	No	Time-Base Status Register
TBCTL		1	No	No	Time-Base Control Register
TBPHS		1	No	No	Time-Base Phase Register
TBPRD		1	Yes	No	Time-Base Period Register
TBCTR		1	No	No	Time-Base Counter Register
Counter-Compare Submodule Registers					
CMPCTL		1	No	No	Counter-Compare Control Register
CMPA		1	Yes	No	Counter-Compare A Register
CMPB		1	Yes	No	Counter-Compare B Register
Action-Qualifier Submodule Registers					
AQCTLA		1	No	No	Action-Qualifier Control Register for Output A (EPWMxA)
AQSFRC		1	No	No	Action-Qualifier Software Force Register
AQCTLB		1	No	No	Action-Qualifier Control Register for Output B (EPWMxB)
AQCSFRC		1	Yes	No	Action-Qualifier Continuous S/W Force Register Set
Dead-Band Generator Submodule Registers					
DBCTL		1	No	No	Dead-Band Generator Control Register
DBFED		1	No	No	Dead-Band Generator Falling Edge Delay Count Register
DBRED		1	No	No	Dead-Band Generator Rising Edge Delay Count Register
Trip-Zone Submodule Registers					
TZDCSEL		1	No	Yes	Trip Zone Digital Compare Select Register
TZSEL		1	No	Yes	Trip-Zone Select Register
TZEINT		1	No	Yes	Trip-Zone Enable Interrupt Register
TZCTL		1	No	Yes	Trip-Zone Control Register
TZCLR		1	No	Yes	Trip-Zone Clear Register
TZFLG		1	No	No	Trip-Zone Flag Register
TZFRC		1	No	Yes	Trip-Zone Force Register
Event-Trigger Submodule Registers					
ETSEL		1	No	No	Event-Trigger Selection Register
ETFLG		1	No	No	Event-Trigger Flag Register
ETPS		1	No	No	Event-Trigger Pre-Scale Register
ETFRC		1	No	No	Event-Trigger Force Register
ETCLR		1	No	No	Event-Trigger Clear Register
PWM-Chopper Submodule Registers					
PCCTL		1	No	No	PWM-Chopper Control Register

(1) Locations not shown are reserved.

Table 20-1. ePWM Module Control and Status Register Set Grouped by Submodule (continued)

Name	Address Offset[1]	Size (x16)	Shadow	Privileged Mode Write Only?	Description
Digital Compare Event Registers					
DCACTL		1	No	Yes	Digital Compare A Control Register
DCTRIPSEL		1	No	Yes	Digital Compare Trip Select Register
DCFCTL		1	No	Yes	Digital Compare Filter Control Register
DCBCTL		1	No	Yes	Digital Compare B Control Register
DCFOFFSET		1	Writes	No	Digital Compare Filter Offset Register
DCCAPCTL		1	No	Yes	Digital Compare Capture Control Register
DCFWINDOW		1	No	No	Digital Compare Filter Window Register
DCFOFFSETCNT		1	No	No	Digital Compare Filter Offset Counter Register
DCCAP		1	Yes	No	Digital Compare Counter Capture Register
DCFWINDOWCNT		1	No	No	Digital Compare Filter Window Counter Register

20.2 ePWM Submodules

Eight submodules are included in every ePWM peripheral. Each of these submodules performs specific tasks that can be configured by software.

20.2.1 Overview

Table 20-2 lists the eight key submodules together with a list of their main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, then you should see the counter-compare submodule in Section 20.2.3 for relevant details.

Table 20-2. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time-base (TB)	• Scale the time-base clock (TBCLK) relative to the system clock (VCLK3). • Configure the PWM time-base counter (TBCTR) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-down mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another ePWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Configure how the time-base counter will behave when the device is halted by an emulator. • Specify the source for the synchronization output of the ePWM module: – Synchronization input signal – Time-base counter equal to zero – Time-base counter equal to counter-compare B (CMPB) – No output synchronization signal generated.
Counter-compare (CC)	• Specify the PWM duty cycle for output EPWMxA and/or output EPWMxB • Specify the time at which switching events occur on the EPWMxA or EPWMxB output
Action-qualifier (AQ)	• Specify the type of action taken when a time-base or counter-compare submodule event occurs: – No action taken – Output EPWMxA and/or EPWMxB switched high – Output EPWMxA and/or EPWMxB switched low – Output EPWMxA and/or EPWMxB toggled • Force the PWM output state through software control • Configure and control the PWM dead-band through software

Table 20-2. Submodule Configuration Parameters (continued)

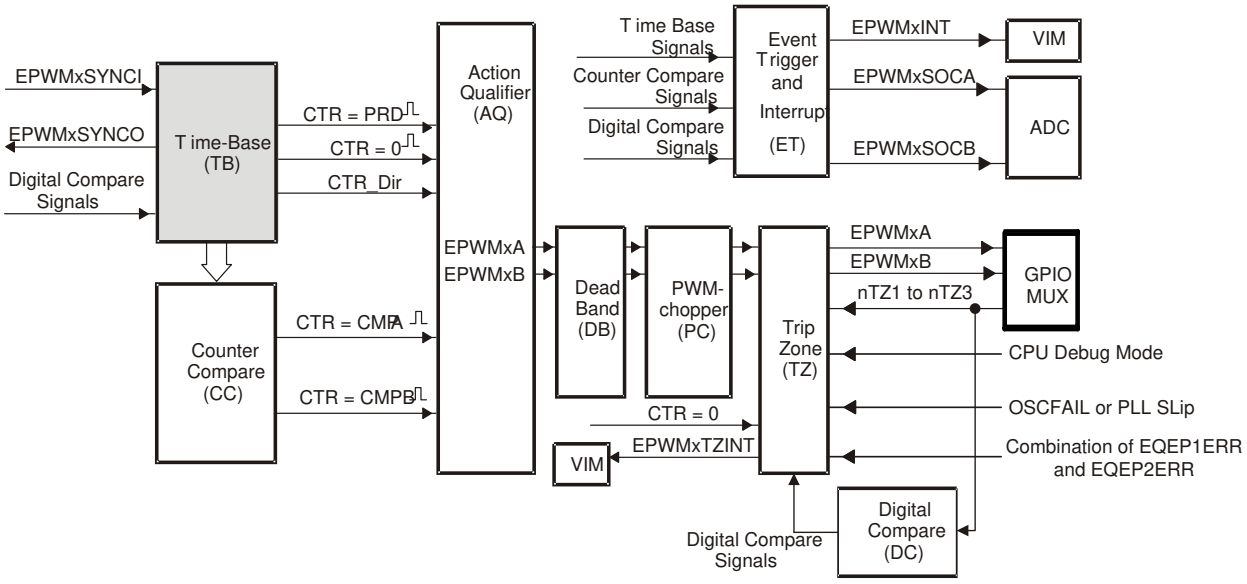
Submodule	Configuration Parameter or Option
Dead-band (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification. • Option to enable half-cycle clocking for double resolution.
PWM-chopper (PC)	• Create a chopping (carrier) frequency. • Pulse width of the first pulse in the chopped pulse train. • Duty cycle of the second and subsequent pulses. • Bypass the PWM-chopper module entirely. In this case the PWM waveform is passed through without modification.
Trip-zone (TZ)	• Configure the ePWM module to react to one, all, or none of the trip-zone signals or digital compare events. • Specify the tripping action taken when a fault occurs: – Force EPWMxA and/or EPWMxB high – Force EPWMxA and/or EPWMxB low – Force EPWMxA and/or EPWMxB to a high-impedance state – Configure EPWMxA and/or EPWMxB to ignore any trip condition. • Configure how often the ePWM will react to each trip-zone signal: – One-shot – Cycle-by-cycle • Enable the trip-zone to initiate an interrupt. • Bypass the trip-zone module entirely.
Event-trigger (ET)	• Enable the ePWM events that will trigger an interrupt. • Enable ePWM events that will trigger an ADC start-of-conversion event. • Specify the rate at which events cause triggers (every occurrence or every second or third occurrence) • Poll, set, or clear event flags
Digital-compare (DC)	• Enables trip zone signals to create events and filtered events • Specify event-filtering options to capture TBCTR counter or generate blanking window

Code examples are provided in this chapter that show how to implement various ePWM module configurations. These examples use the constant definitions in the device *EPwm_defines.h* file in the device-specific header file and peripheral examples software package.

20.2.2 Time-Base (TB) Submodule

Each ePWM module has its own time-base submodule that determines all of the event timing for the ePWM module. Built-in synchronization logic allows the time-base of multiple ePWM modules to work together as a single system. Figure 20-2 illustrates the time-base module's place within the ePWM.

Figure 20-2. Time-Base Submodule Block Diagram



20.2.2.1 Purpose of the Time-Base Submodule

You can configure the time-base submodule for the following:

- Specify the ePWM time-base counter (TBCTR) frequency or period to control how often events occur.
- Manage time-base synchronization with other ePWM modules.
- Maintain a phase relationship with other ePWM modules.
- Set the time-base counter to count-up, count-down, or count-up-and-down mode.
- Generate the following events:
  - CTR = PRD: Time-base counter equal to the specified period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000).
- Configure the rate of the time-base clock; a prescaled version of the device peripheral clock domain (VCLK3). This allows the time-base counter to increment/decrement at a slower rate.

20.2.2.2 Controlling and Monitoring the Time-Base Submodule

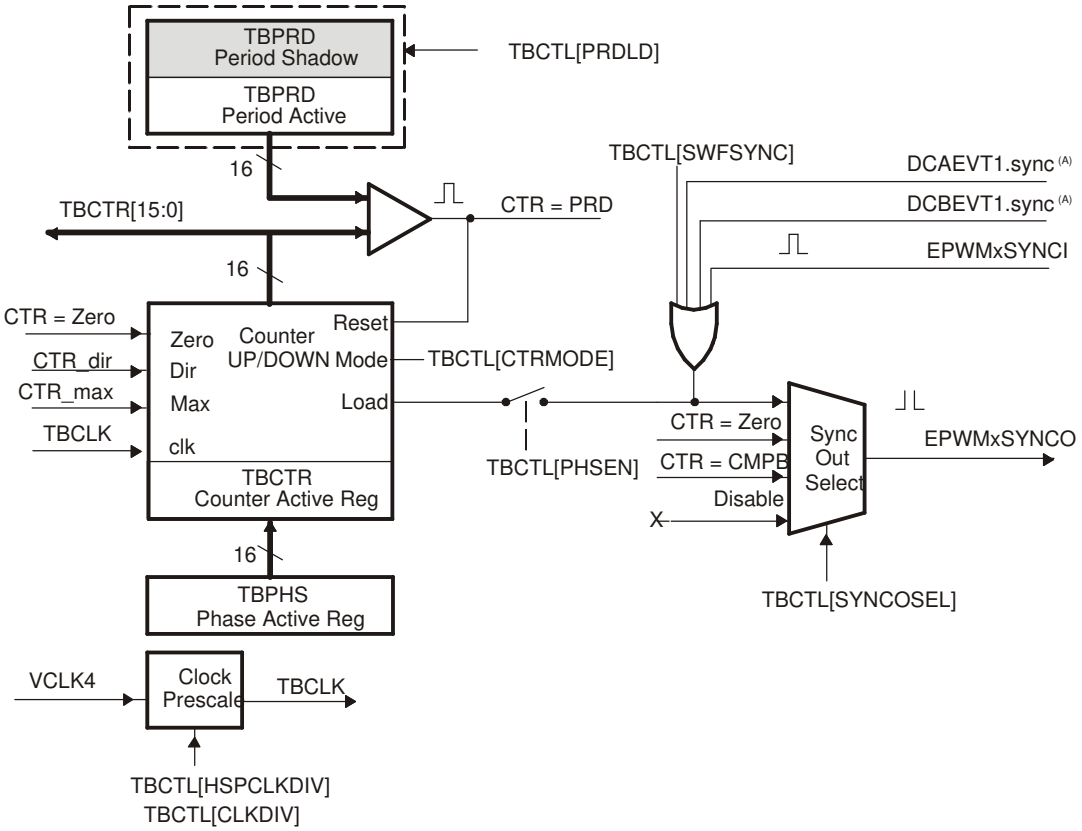
Table 20-3 shows the registers used to control and monitor the time-base submodule.

Table 20-3. Time-Base Submodule Registers

Register	Address Offset	Shadowed	Description
TBSTS		No	Time-Base Status Register
TBCTL		No	Time-Base Control Register
TBPHS		No	Time-Base Phase Register
TBPRD		Yes	Time-Base Period Register
TBCTR		No	Time-Base Counter Register

Figure 20-3 shows the critical signals and registers of the time-base submodule. Table 20-4 provides descriptions of the key signals associated with the time-base submodule.

Figure 20-3. Time-Base Submodule Signals and Registers



A. These signals are generated by the digital compare (DC) submodule.

Table 20-4. Key Time-Base Signals

Signal	Description
EPWMxSYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of ePWM module earlier in the synchronization chain. An ePWM peripheral can be configured to use or ignore this signal. For the first ePWM module (EPWM1) this signal comes from a device pin or from the N2HET1 module. For subsequent ePWM modules this signal is passed from another ePWM peripheral. For example, EPWM2SYNCI is generated by the ePWM1 peripheral, EPWM3SYNCI is generated by ePWM2 and so forth. See Section 20.2.2.3.3 for information on the synchronization order of a particular device.
EPWMxSYNCO	Time-base synchronization output. This output pulse is used to synchronize the counter of an ePWM module later in the synchronization chain. The ePWM module generates this signal from one of three event sources: 1. EPWMxSYNCI (Synchronization input pulse) 2. CTR = Zero: The time-base counter equal to zero (TBCTR = 0x0000). 3. CTR = CMPB: The time-base counter equal to the counter-compare B (TBCTR = CMPB) register.
CTR = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCTR = TBPRD.
CTR = Zero	Time-base counter equal to zero This signal is generated whenever the counter value is zero. That is when TBCTR equals 0x0000.

Table 20-4. Key Time-Base Signals (continued)

Signal	Description
CTR = CMPB	Time-base counter equal to active counter-compare B register (TBCTR = CMPB).
	This event is generated by the counter-compare submodule and used by the synchronization out logic
CTR_dir	Time-base counter direction.
	Indicates the current direction of the ePWM's time-base counter. This signal is high when the counter is increasing and low when it is decreasing.
CTR_max	Time-base counter equal max value. (TBCTR = 0xFFFF)
	Generated event when the TBCTR value reaches its maximum value. This signal is only used only as a status bit
TBCLK	Time-base clock.
	This is a prescaled version of the system clock (VCLK3) and is used by all submodules within the ePWM. This clock determines the rate at which time-base counter increments or decrements.

20.2.2.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (TBPRD) register and the mode of the time-base counter. Figure 20-4 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count, down-count, and up-down-count time-base counter modes when the period is set to 4 (TBPRD = 4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the system clock (VCLK3).

The time-base counter has three modes of operation selected by the time-base control register (TBCTL):

- **Up-Down-Count Mode:**

  In up-down-count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until it reaches zero. At this point the counter repeats the pattern and begins to increment.

- **Up-Count Mode:**

  In this mode, the time-base counter starts from zero and increments until it reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.

- **Down-Count Mode:**

  In down-count mode, the time-base counter starts from the period (TBPRD) value and decrements until it reaches zero. When it reaches zero, the time-base counter is reset to the period value and it begins to decrement once again.

20.2.2.3.1 Time-Base Period Shadow Register

The time-base period register (TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the ePWM module:

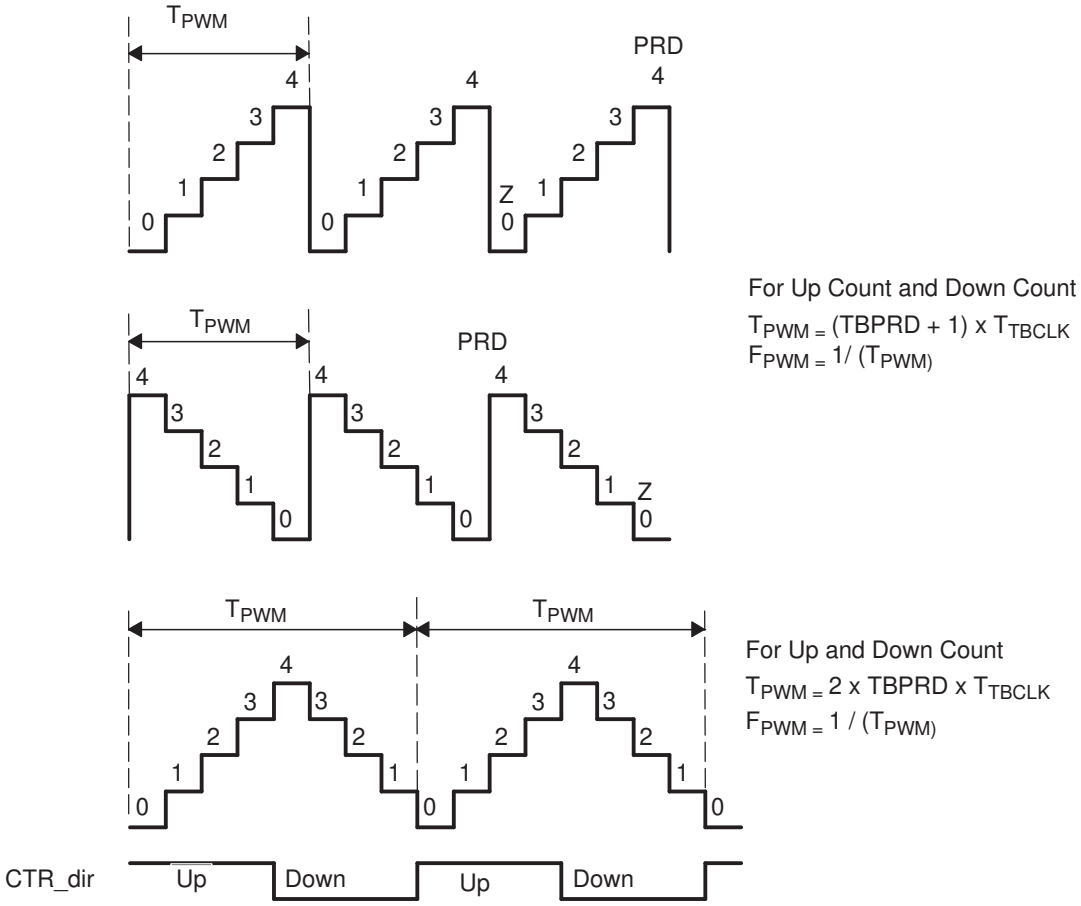
- **Active Register**

  The active register controls the hardware and is responsible for actions that the hardware causes or invokes.

- **Shadow Register**

  The shadow register buffers or provides a temporary holding location for the active register. It has no direct effect on any control hardware. At a strategic point in time the shadow register's content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

Figure 20-4. Time-Base Frequency and Period



The memory address of the shadow period register is the same as the active register. Which register is written to or read from is determined by the TBCTL[PRDLD] bit. This bit enables and disables the TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:**

  The TBPRD shadow register is enabled when TBCTL[PRDLD] = 0. Reads from and writes to the TBPRD memory address go to the shadow register. The shadow register contents are transferred to the active register (TBPRD (Active) ← TBPRD (shadow)) when the time-base counter equals zero (TBCTR = 0x0000). By default the TBPRD shadow register is enabled.
- **Time-Base Period Immediate Load Mode:**

  If immediate load mode is selected (TBCTL[PRDLD] = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

20.2.2.3.2 Time-Base Clock Synchronization

Bit 1 of the device-level multiplexing control module (IOMM) register PINMMR166 is defined as the TBCLKSYNC bit. The TBCLKSYNC bit allows users to globally synchronize all enabled ePWM modules to the time-base clock (TBCLK). When set, all enabled ePWM module clocks are started with the first rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescalers for each ePWM module must be set identically.

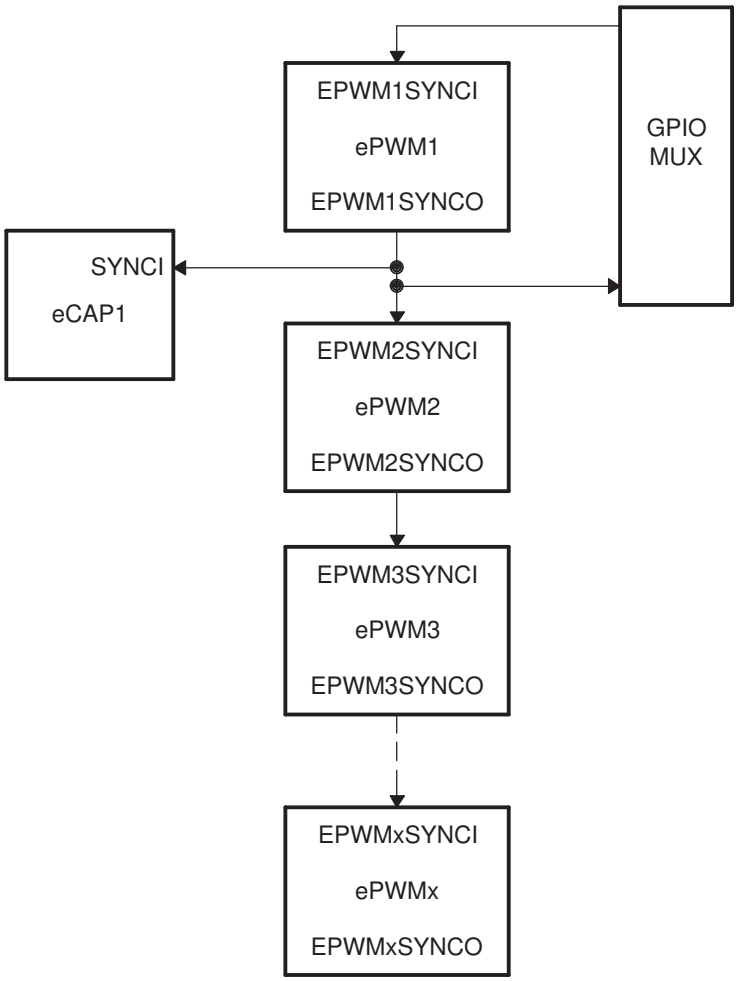
The proper procedure for enabling ePWM clocks is as follows:

1. Enable ePWM module clocks using the IOMM control registers for each ePWM module instance
2. Set TBCLKSYNC = 0. This will stop the time-base clock within any enabled ePWM module.
3. Configure ePWM modules: prescaler values and ePWM modes.
4. Set TBCLKSYNC = 1.

20.2.2.3.3 Time-Base Counter Synchronization

A time-base synchronization scheme connects all of the ePWM modules on a device. Each ePWM module has a synchronization input (EPWMxSYNCI) and a synchronization output (EPWMxSYNCO). The input synchronization for the first instance (ePWM1) comes from an external pin. The synchronization connections for the remaining ePWM modules are shown in Figure 20-5.

Figure 20-5. Time-Base Counter Synchronization Scheme



Each ePWM module can be configured to use or ignore the synchronization input. If the TBCTL[PHSEN] bit is set, then the time-base counter (TBCTR) of the ePWM module will be automatically loaded with the phase register (TBPHS) contents when one of the following conditions occur:

- **EPWMxSYNCI: Synchronization Input Pulse:**

  The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (TBPHS → TBCTR). This operation occurs on the next valid time-base clock (TBCLK) edge.

  The delay from internal master module to slave modules is given by:

  – if (TBCLK = VCLK3): 2 x VCLK3
  – if (TBCLK != VCLK3): 1 TBCLK

- **Software Forced Synchronization Pulse:**

  Writing a 1 to the TBCTL[SWFSYNC] control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on EPWMxSYNCI.

- **Digital Compare Event Synchronization Pulse:**

  DCAEVT1 and DCBEVT1 digital compare events can be configured to generate synchronization pulses which have the same affect as EPWMxSYNCI.

This feature enables the ePWM module to be automatically synchronized to the time base of another ePWM module. Lead or lag phase control can be added to the waveforms generated by different ePWM modules to synchronize them. In up-down-count mode, the TBCTL[PSHDIR] bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The PHSDIR bit is ignored in count-up or count-down modes. See Figure 20-6 through Figure 20-9 for examples.

Clearing the TBCTL[PHSEN] bit configures the ePWM to ignore the synchronization input pulse. The synchronization pulse can still be allowed to flow-through to the EPWMxSYNCO and be used to synchronize other ePWM modules. In this way, you can set up a master time-base (for example, ePWM1) and downstream modules (ePWM2 - ePWMx) may elect to run in synchronization with the master.

20.2.2.4 Phase Locking the Time-Base Clocks of Multiple ePWM Modules

The TBCLKSYNC bit can be used to globally synchronize the time-base clocks of all enabled ePWM modules on a device. When TBCLKSYNC = 0, the time-base clock of all ePWM modules is stopped (default). When TBCLKSYNC = 1, all ePWM time-base clocks are started with the rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescaler bits in the TBCTL register of each ePWM module must be set identically. The proper procedure for enabling the ePWM clocks is as follows:

1. Enable ePWM module clocks using the IOMM control registers for each ePWM module instance
2. Set TBCLKSYNC= 0. This will stop the time-base clock within any enabled ePWM module.
3. Configure ePWM modules: prescaler values and ePWM modes.
4. Set TBCLKSYNC=1.

20.2.2.5 Time-base Counter Modes and Timing Waveforms

The time-base counter operates in one of four modes:

- Up-count mode which is asymmetrical.
- Down-count mode which is asymmetrical.
- Up-down-count which is symmetrical
- Frozen where the time-base counter is held constant at the current value

To illustrate the operation of the first three modes, the following timing diagrams show when events are generated and how the time-base responds to an EPWMxSYNCI signal.

Figure 20-6. Time-Base Up-Count Mode Waveforms

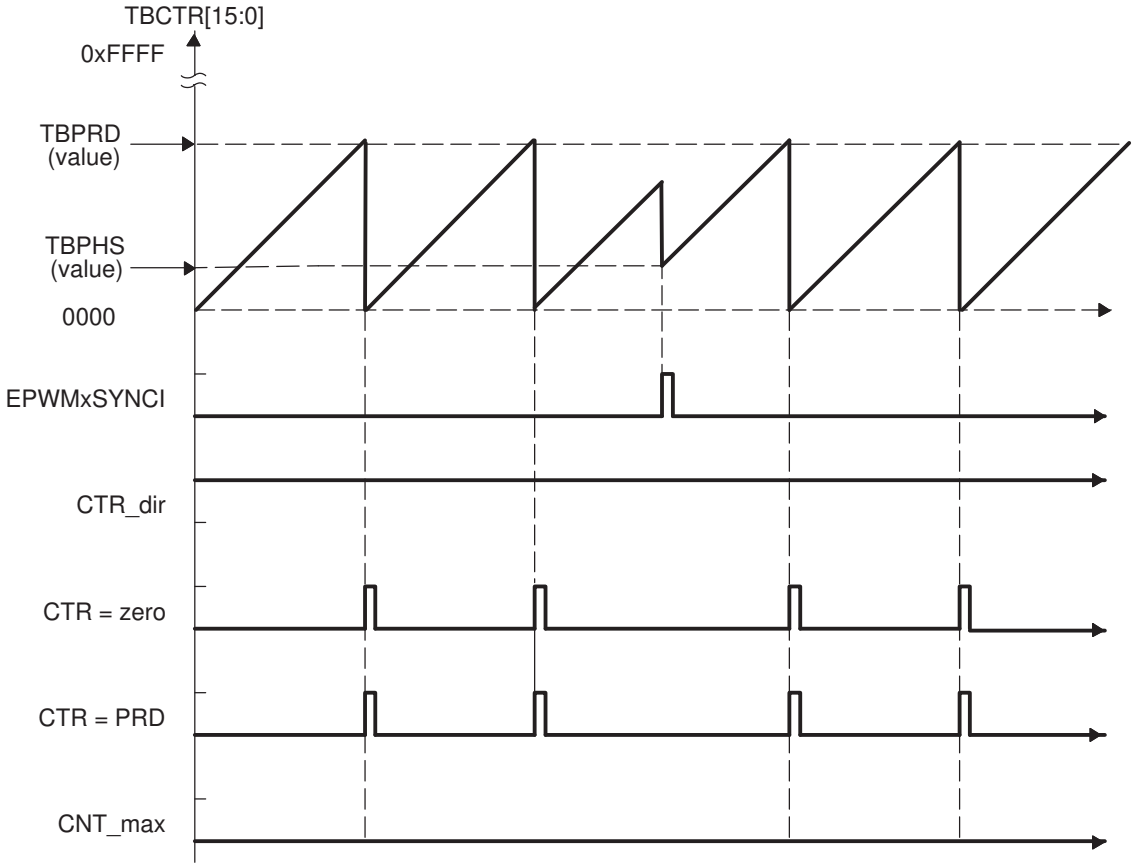


Figure 20-7. Time-Base Down-Count Mode Waveforms

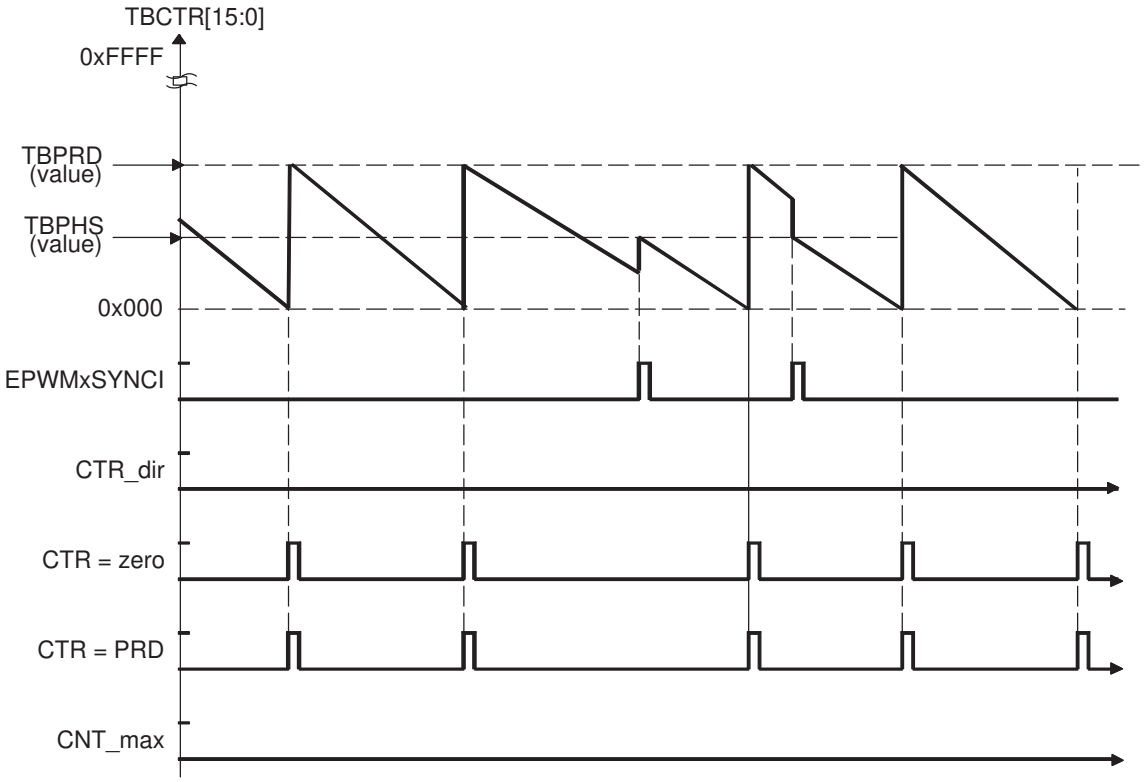


Figure 20-8. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

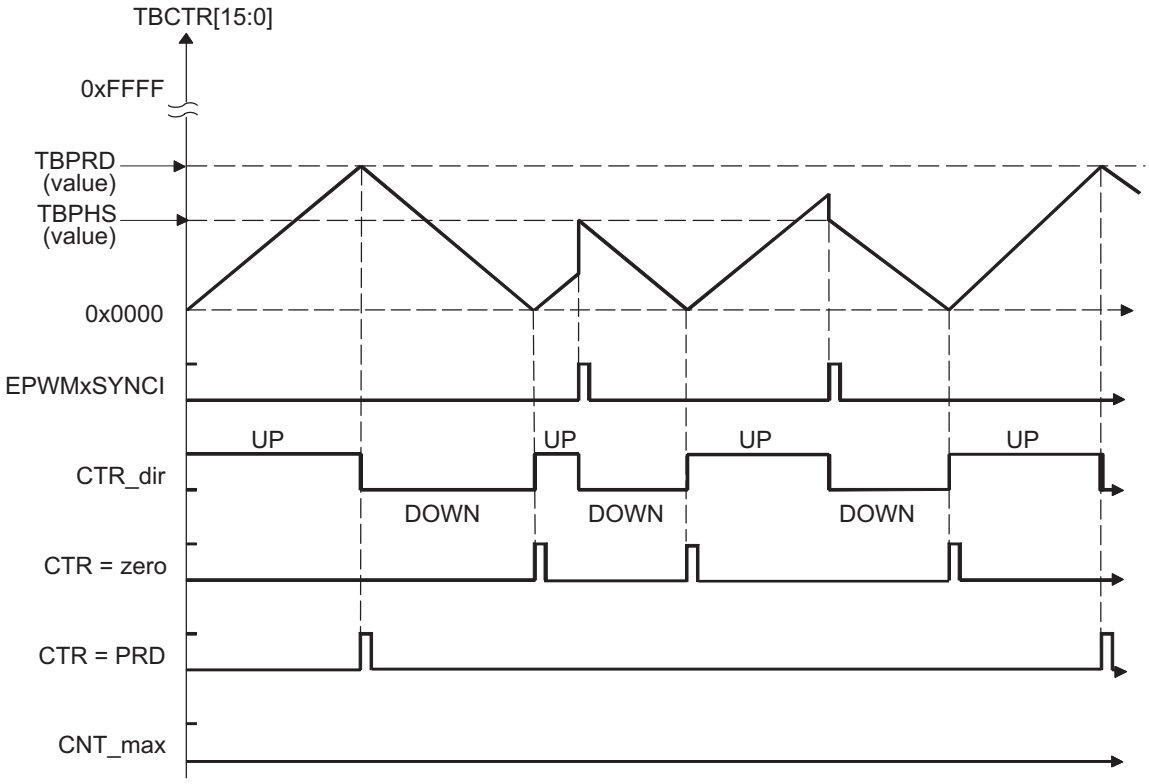
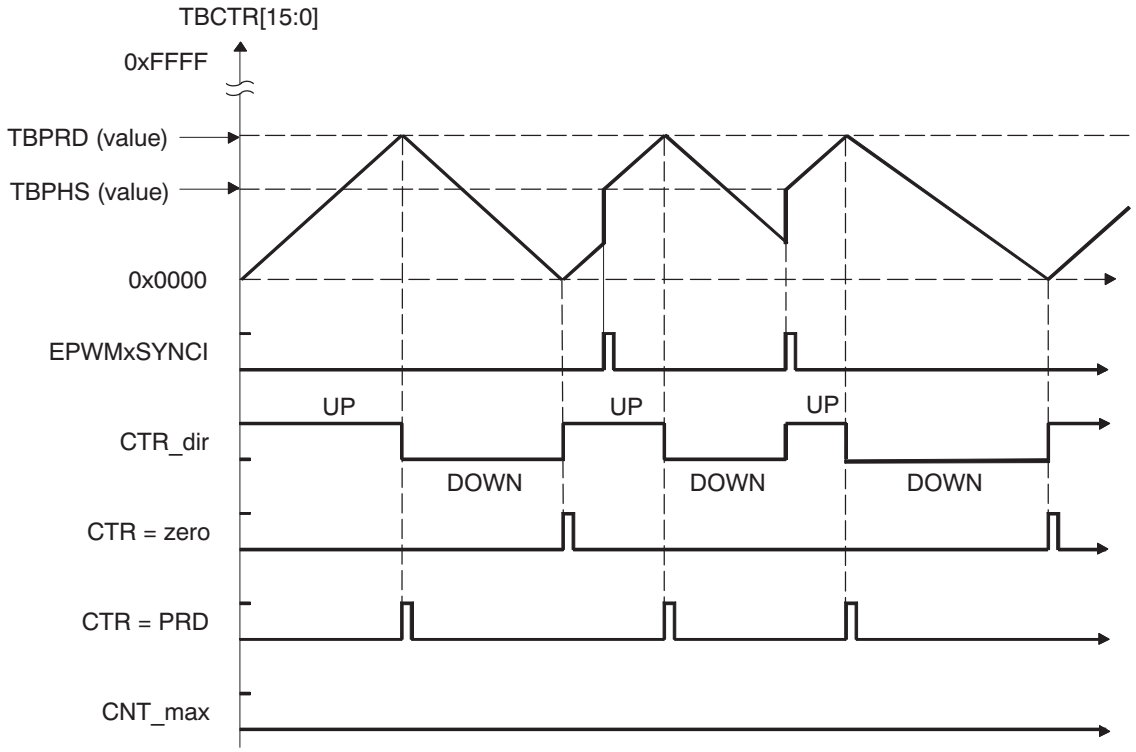


Figure 20-9. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event



20.2.3 Counter-Compare (CC) Submodule

Figure 20-10 illustrates the counter-compare submodule within the ePWM.

Figure 20-11 shows the basic structure of the counter-compare submodule.

Figure 20-10. Counter-Compare Submodule

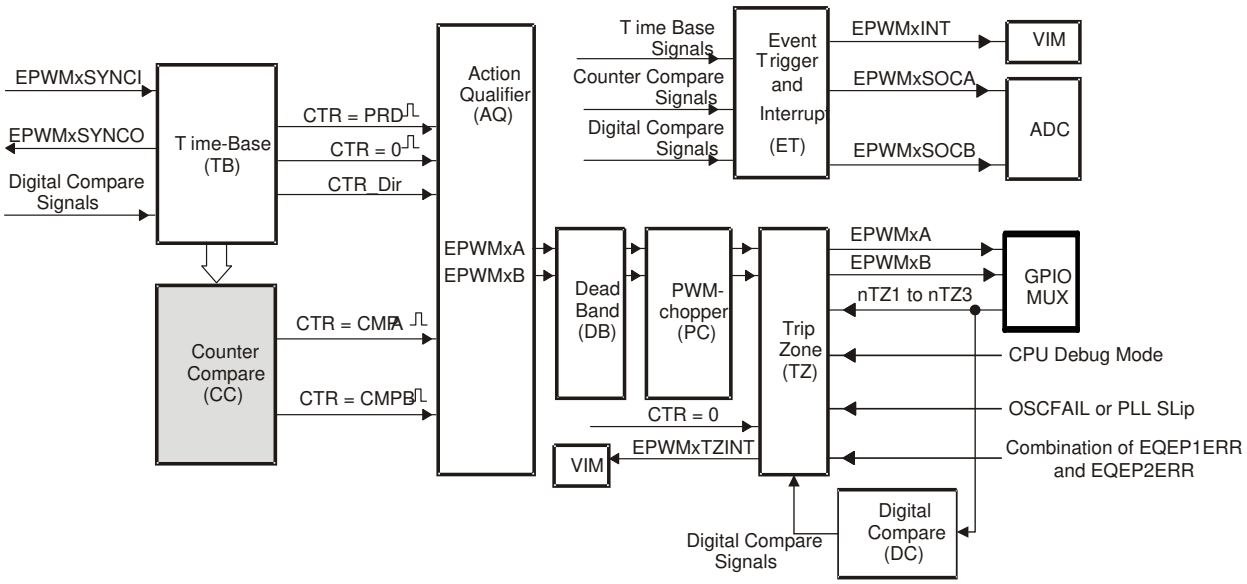
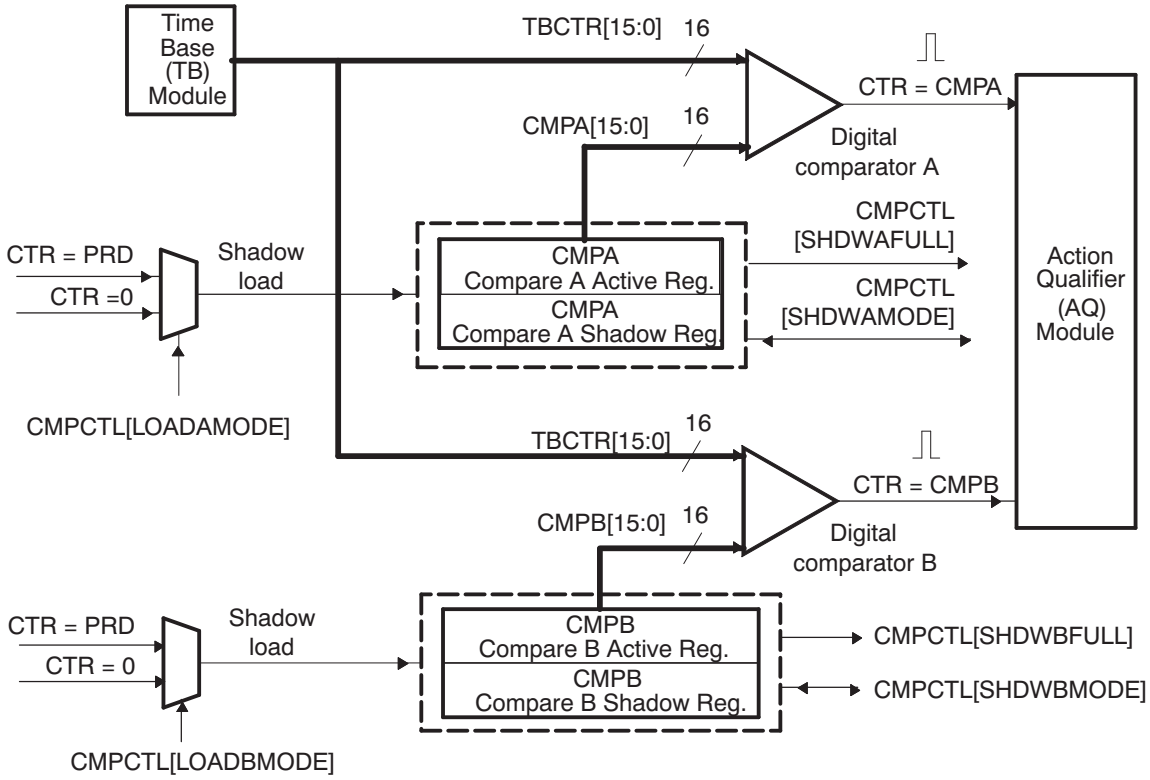


Figure 20-11. Detailed View of the Counter-Compare Submodule



20.2.3.1 Purpose of the Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (CMPA) and counter-compare B (CMPB) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event.

The counter-compare:

- Generates events based on programmable time stamps using the CMPA and CMPB registers
  - CTR = CMPA: Time-base counter equals counter-compare A register (TBCTR = CMPA).
  - CTR = CMPB: Time-base counter equals counter-compare B register (TBCTR = CMPB)
- Controls the PWM duty cycle if the action-qualifier submodule is configured appropriately
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

20.2.3.2 Controlling and Monitoring the Counter-Compare Submodule

The counter-compare submodule operation is controlled and monitored by the registers listed in Table 20-5.

The key signals associated with the counter-compare submodule are described in Table 20-6.

Table 20-5. Counter-Compare Submodule Registers

Register Name	Address Offset	Shadowed	Description
CMPCTL		No	Counter-Compare Control Register.
CMPA		Yes	Counter-Compare A Register
CMPB		Yes	Counter-Compare B Register

Table 20-6. Counter-Compare Submodule Key Signals

Signal	Description of Event	Registers Compared
CTR = CMPA	Time-base counter equal to the active counter-compare A value	TBCTR = CMPA
CTR = CMPB	Time-base counter equal to the active counter-compare B value	TBCTR = CMPB
CTR = PRD	Time-base counter equal to the active period. Used to load active counter-compare A and B registers from the shadow register	TBCTR = TBPRD
CTR = ZERO	Time-base counter equal to zero. Used to load active counter-compare A and B registers from the shadow register	TBCTR = 0x0000

20.2.3.3 Operational Highlights for the Counter-Compare Submodule

The counter-compare submodule is responsible for generating two independent compare events based on two compare registers:

1. CTR = CMPA: Time-base counter equal to counter-compare A register (TBCTR = CMPA).
2. CTR = CMPB: Time-base counter equal to counter-compare B register (TBCTR = CMPB).

For up-count or down-count mode, each event occurs only once per cycle. For up-down-count mode each event occurs twice per cycle if the compare value is between 0x0000-TBPRD and once per cycle if the compare value is equal to 0x0000 or equal to TBPRD. These events are fed into the action-qualifier submodule where they are qualified by the counter direction and converted into actions if enabled. Refer to Section 20.2.4.1 for more details.

The counter-compare registers CMPA and CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occur at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. The memory address of the active register and the shadow register is identical. Which register is written to or read from is determined by the CMPCTL[SHDWAMODE] and CMPCTL[SHDWBMODE] bits. These bits enable and disable the CMPA shadow register and CMPB shadow register respectively. The behavior of the two load modes is described below:

Shadow Mode:

The shadow mode for the CMPA is enabled by clearing the CMPCTL[SHDWAMODE] bit and the shadow register for CMPB is enabled by clearing the CMPCTL[SHDWBMODE] bit. Shadow mode is enabled by default for both CMPA and CMPB.

If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events as specified by the CMPCTL[LOADAMODE] and CMPCTL[LOADBMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000)
- Both CTR = PRD and CTR = Zero

Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.

Immediate Load Mode:

If immediate load mode is selected (TBCTL[SHADWAMODE] = 1 or TBCTL[SHADWBMODE] = 1), then a read from or a write to the register will go directly to the active register.

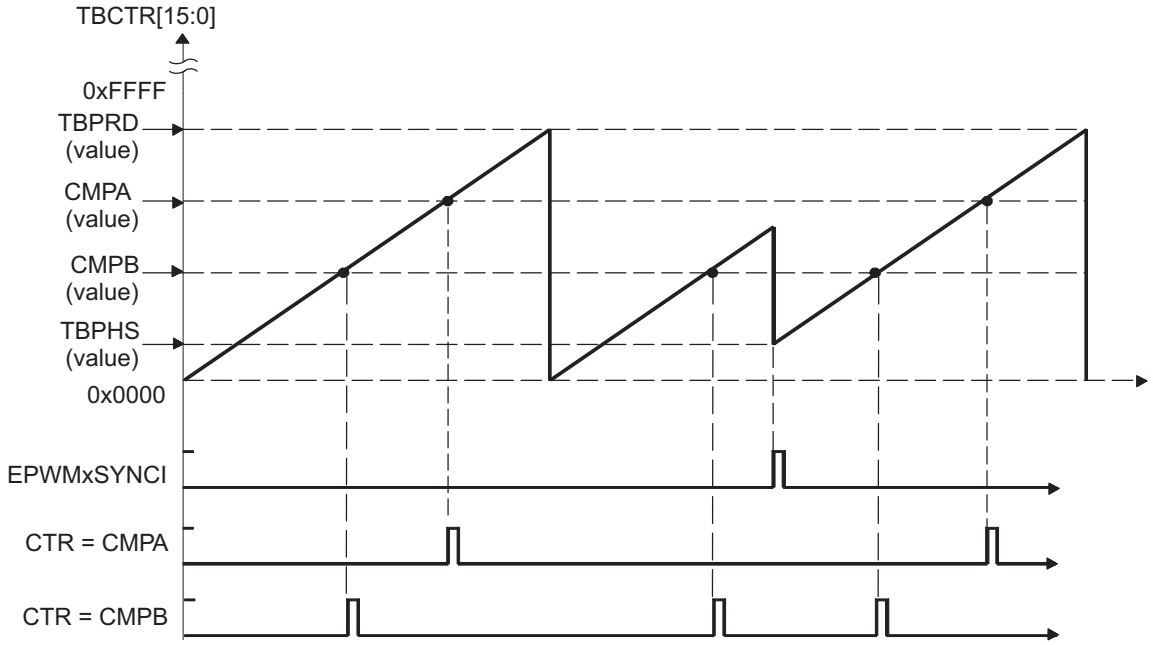
20.2.3.4 Count Mode Timing Waveforms

The counter-compare module can generate compare events in all three count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Down-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To illustrate the operation of the first three modes, the timing diagrams in Figure 20-12 through Figure 20-15 show when events are generated and how the EPWMxSYNCI signal interacts.

Figure 20-12. Counter-Compare Event Waveforms in Up-Count Mode



NOTE: An EPWMxSYNCI external synchronization event can cause a discontinuity in the TBCTR count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 20-13. Counter-Compare Events in Down-Count Mode

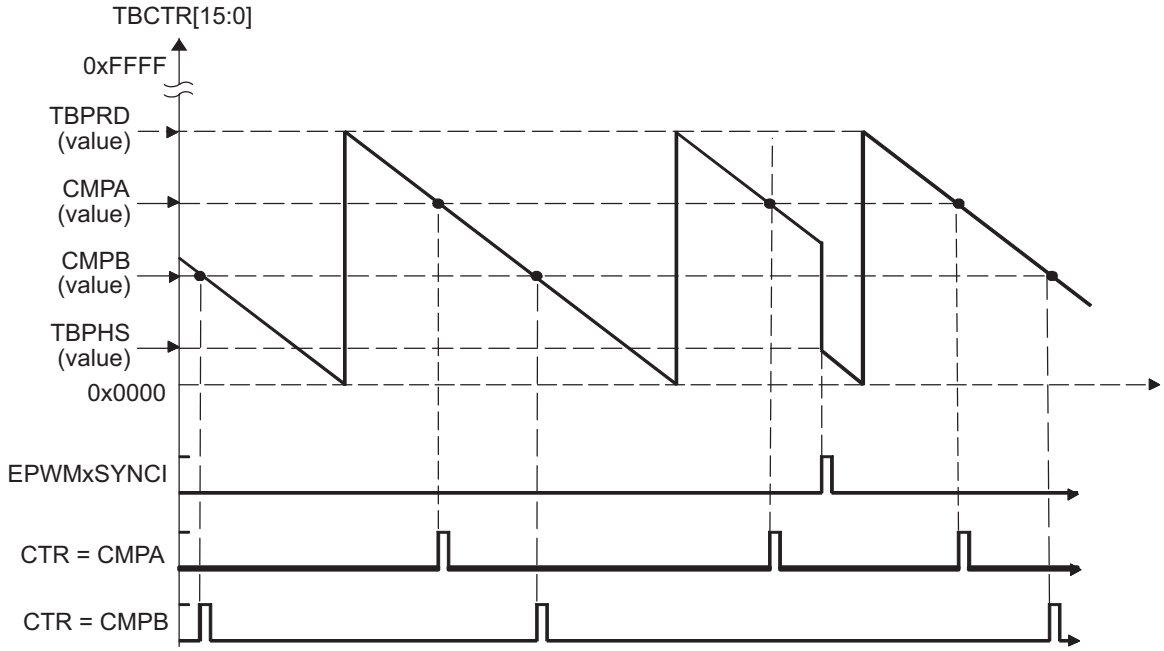


Figure 20-14. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

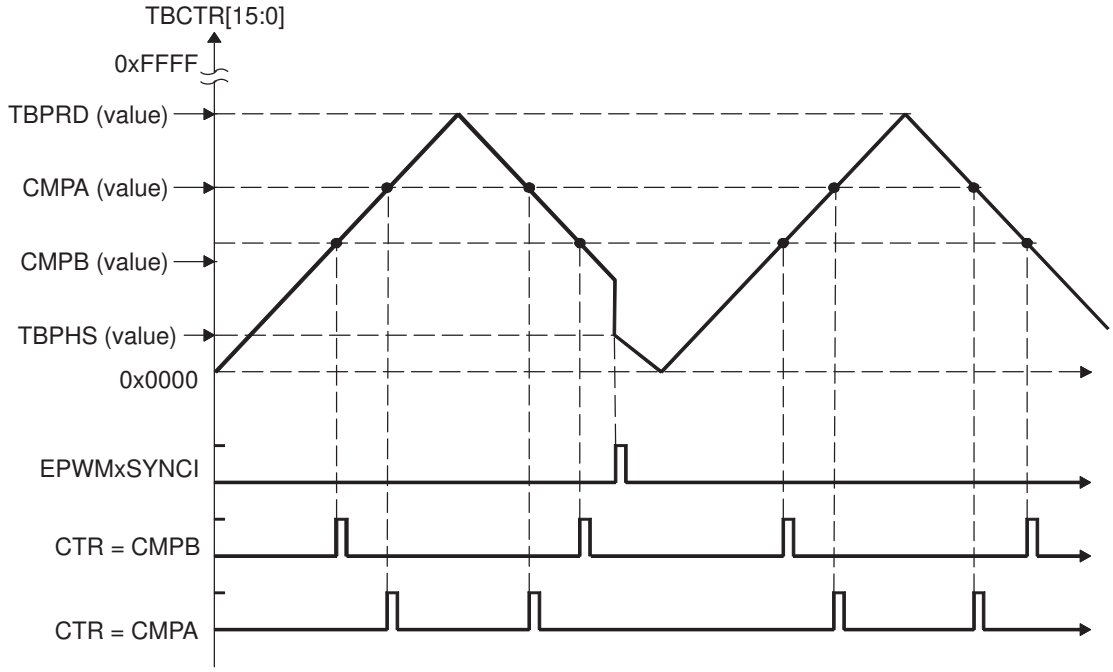
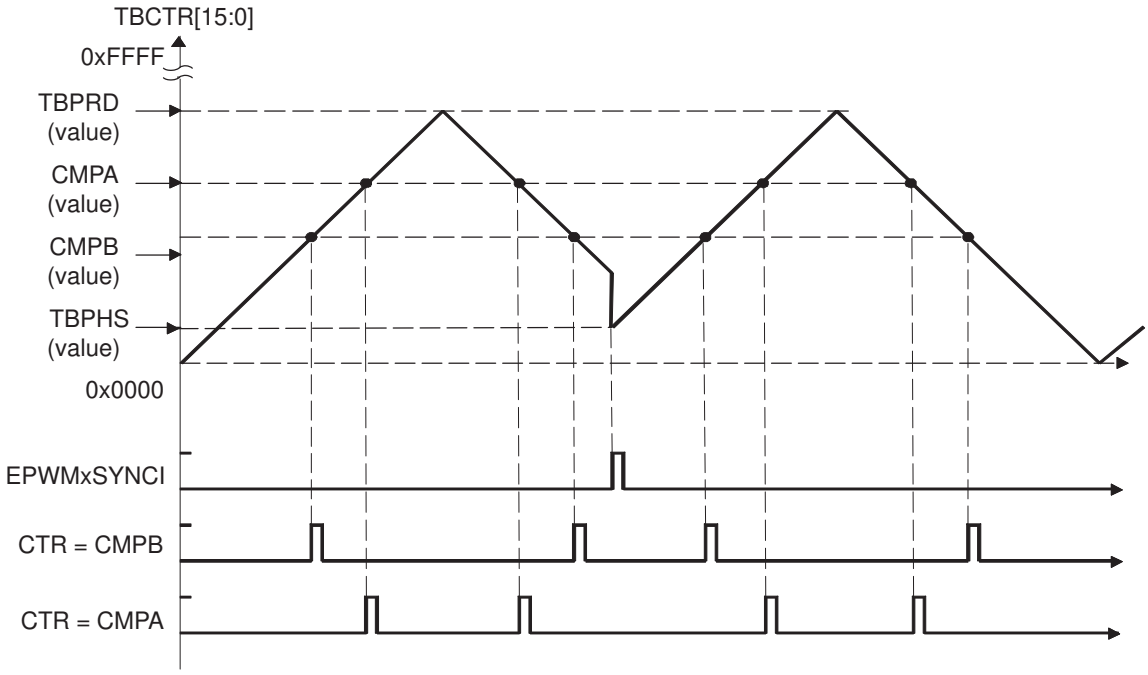


Figure 20-15. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

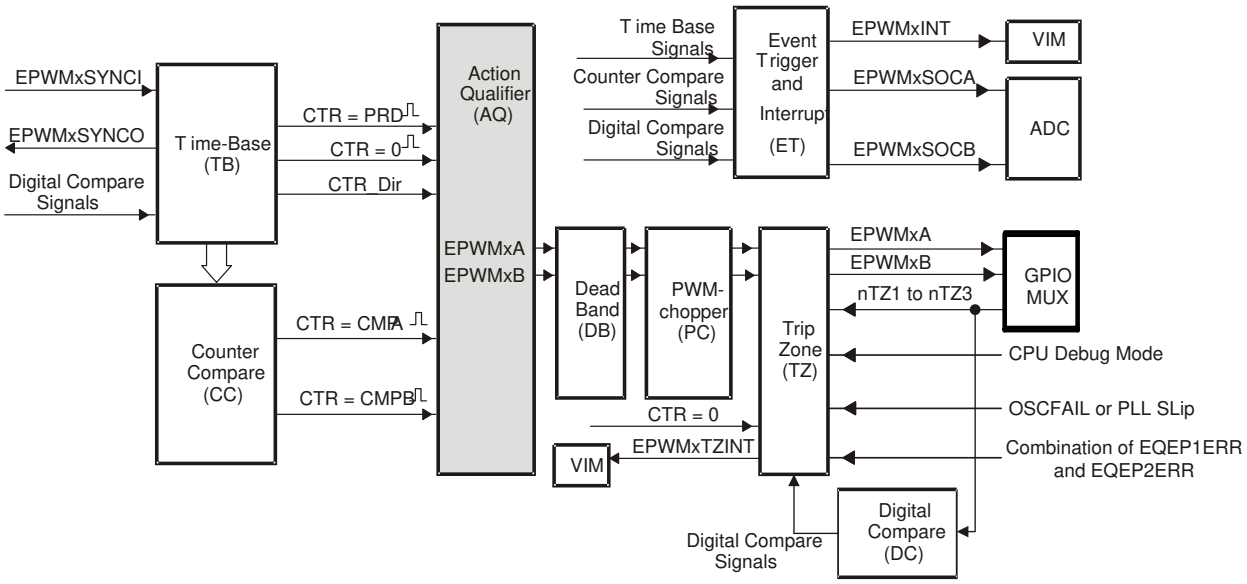


20.2.4 Action-Qualifier (AQ) Submodule

Figure 20-16 shows the action-qualifier (AQ) submodule (see shaded block) in the ePWM system.

The action-qualifier submodule has the most important role in waveform construction and PWM generation. It decides which events are converted into various action types, thereby producing the required switched waveforms at the EPWMxA and EPWMxB outputs.

Figure 20-16. Action-Qualifier Submodule



20.2.4.1 Purpose of the Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000)
  - CTR = CMPA: Time-base counter equal to the counter-compare A register (TBCTR = CMPA)
  - CTR = CMPB: Time-base counter equal to the counter-compare B register (TBCTR = CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when it is decreasing.

20.2.4.2 Action-Qualifier Submodule Control and Status Register Definitions

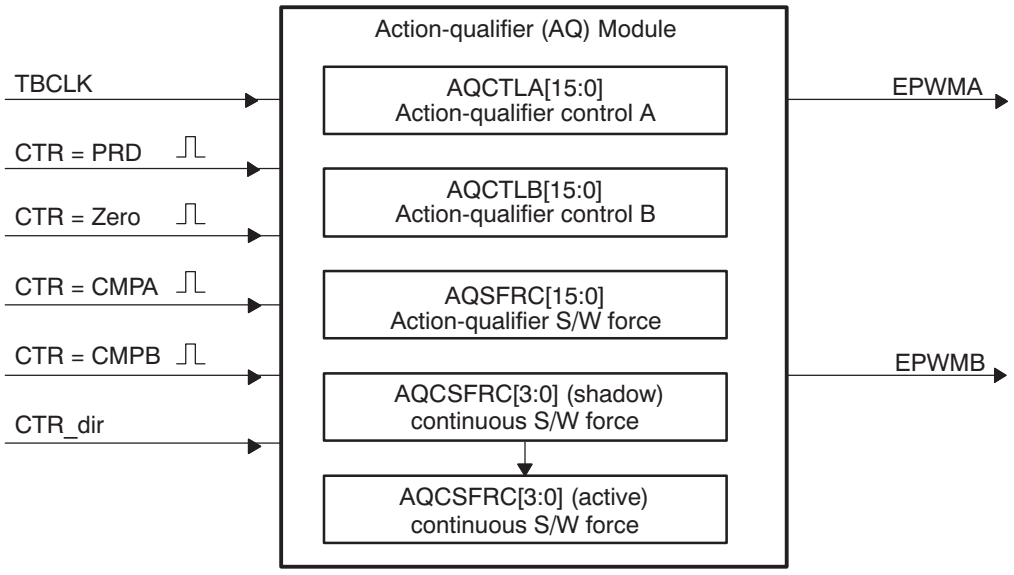
The action-qualifier submodule operation is controlled and monitored via the registers in Table 20-7.

Table 20-7. Action-Qualifier Submodule Registers

Register Name	Address Offset	Shadowed	Description
AQCTLA		No	Action-Qualifier Control Register For Output A (EPWMxA)
AQSFRC		No	Action-Qualifier Software Force Register
AQCTLB		No	Action-Qualifier Control Register For Output B (EPWMxB)
AQCSFRC		Yes	Action-Qualifier Continuous Software Force

The action-qualifier submodule is based on event-driven logic. It can be thought of as a programmable cross switch with events at the input and actions at the output, all of which are software controlled via the set of registers shown in Table 20-7.

Figure 20-17. Action-Qualifier Submodule Inputs and Outputs



The possible input events are summarized again in Table 20-8.

Table 20-8. Action-Qualifier Submodule Possible Input Events

Signal	Description	Registers Compared
CTR = PRD	Time-base counter equal to the period value	TBCTR = TBPRD
CTR = Zero	Time-base counter equal to zero	TBCTR = 0x0000
CTR = CMPA	Time-base counter equal to the counter-compare A	TBCTR = CMPA
CTR = CMPB	Time-base counter equal to the counter-compare B	TBCTR = CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by registers AQSFRC and AQCSFRC.

The action-qualifier submodule controls how the two outputs EPWMxA and EPWMxB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs EPWMxA and EPWMxB are:

- **Set High:**

  Set output EPWMxA or EPWMxB to a high level.
- **Clear Low:**

  Set output EPWMxA or EPWMxB to a low level.
- **Toggle:**

  If EPWMxA or EPWMxB is currently pulled high, then pull the output low. If EPWMxA or EPWMxB is currently pulled low, then pull the output high.

- **Do Nothing:**

  Keep outputs EPWMxA and EPWMxB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the EPWMxA and EPWMxB outputs, this event can still trigger interrupts and ADC start of conversion. See the Event-trigger Submodule description in Section 20.2.8 for details.

Actions are specified independently for either output (EPWMxA or EPWMxB). Any or all events can be configured to generate actions on a given output. For example, both CTR = CMPA and CTR = CMPB can operate on output EPWMxA. All qualifier actions are configured via the control registers found at the end of this section.

For clarity, the drawings in this document use a set of symbolic actions. These symbols are summarized in Figure 20-18. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and their time positions are programmed via the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"; it is the default at reset.

Figure 20-18. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs

S/W force	TB Counter equals:				Actions
	Zero	Comp A	Comp B	Period	
SW ✕	Z ✕	CA ✕	CB ✕	P ✕	Do Nothing
SW ↓	Z ↓	CA ↓	CB ↓	P ↓	Clear Low
SW ↑	Z ↑	CA ↑	CB ↑	P ↑	Set High
SW T	Z T	CA T	CB T	P T	Toggle

20.2.4.3 Action-Qualifier Event Priority

It is possible for the ePWM action qualifier to receive more than one event at the same time. In this case events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down-count mode are shown in Table 20-9. A priority level of 1 is the highest priority and level 7 is the lowest. The priority changes slightly depending on the direction of TBCTR.

Table 20-9. Action-Qualifier Event Priority for Up-Down-Count Mode

Priority Level	Event If TBCTR is Incrementing TBCTR = Zero up to TBCTR = TBPRD	Event If TBCTR is Decrementing TBCTR = TBPRD down to TBCTR = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4	Counter equals zero	Counter equals period (TBPRD)
5	Counter equals CMPB on down-count (CBD)	Counter equals CMPB on up-count (CBU)
6 (Lowest)	Counter equals CMPA on down-count (CAD)	Counter equals CMPA on up-count (CBU)

Table 20-10 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up and thus down-count events will never be taken.

Table 20-10. Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

Table 20-11 shows the action-qualifier priority for down-count mode. In this case, the counter direction is always defined as down and thus up-count events will never be taken.

Table 20-11. Action-Qualifier Event Priority for Down-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to Zero
3	Counter equal to CMPB on down-count (CBD)
4	Counter equal to CMPA on down-count (CAD)
5 (Lowest)	Counter equal to period (TBPRD)

It is possible to set the compare value greater than the period. In this case the action will take place as shown in Table 20-12.

Table 20-12. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAD/CBD	Compare on Down-Count Event CAD/CBD
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event will not occur.	Never occurs.

Table 20-12. Behavior if CMPA/CMPB is Greater than the Period (continued)

Counter Mode	Compare on Up-Count Event CAD/CBD	Compare on Down-Count Event CAD/CBD
Down-Count Mode	Never occurs.	If CMPA/CMPB < TBPRD, the event will occur on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event will occur on a period match (TBCTR=TBPRD).
Up-Down-Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event will occur on a period match (TBCTR = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCTR=CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCTR=TBPRD).

20.2.4.4 Waveforms for Common Configurations

NOTE: The waveforms in this document show the ePWMs behavior for a static compare register value. In a running system, the active compare registers (CMPA and CMPB) are typically updated from their respective shadow registers once every period. The user specifies when the update will take place; either when the time-base counter reaches zero or when the time-base counter reaches period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down-count mode to generate a symmetric PWM:

- If you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1.
- If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1.

  This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down-count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM use the following configuration: Load CMPA/CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

When using up-count mode to generate an asymmetric PWM:

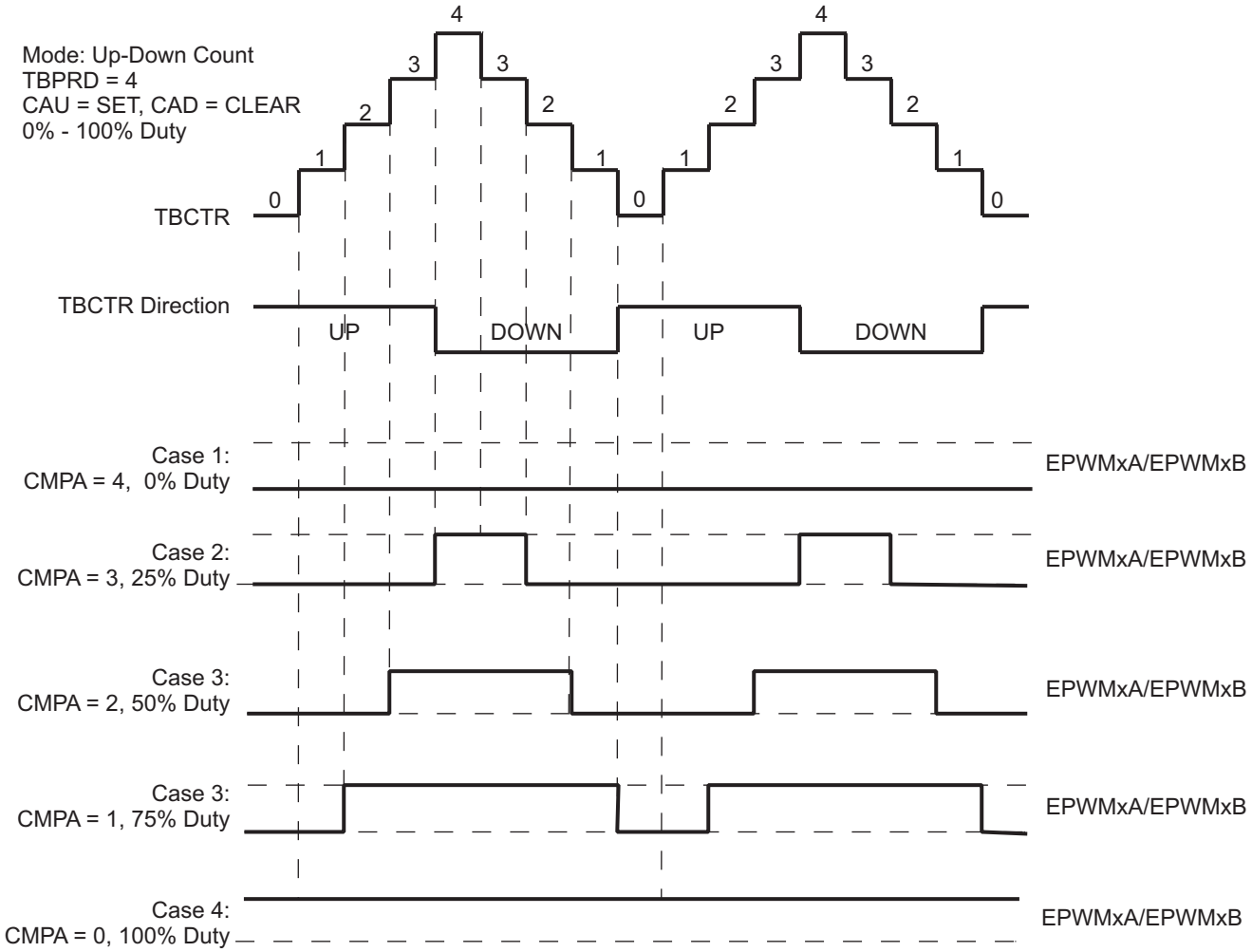
- To achieve 0-100% asymmetric PWM use the following configuration: Load CMPA/CMPB on TBPRD. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD+1 to achieve 0-100% PWM duty.

See the *Using Enhanced Pulse Width Modulator (ePWM) Module for 0-100% Duty Cycle Control* Application Report (literature number SPRAAI1)

Figure 20-19 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCTR. In this mode 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, CMPA is used to make the comparison. When the counter is incrementing the CMPA match will pull the PWM output high. Likewise, when the counter is decrementing the compare match will pull the PWM signal low. When CMPA = 0, the PWM signal is low for the entire period giving the 0% duty waveform. When CMPA = TBPRD, the PWM signal is high achieving 100% duty.

When using this configuration in practice, if you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1. If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1. This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Figure 20-19. Up-Down-Count Mode Symmetrical Waveform

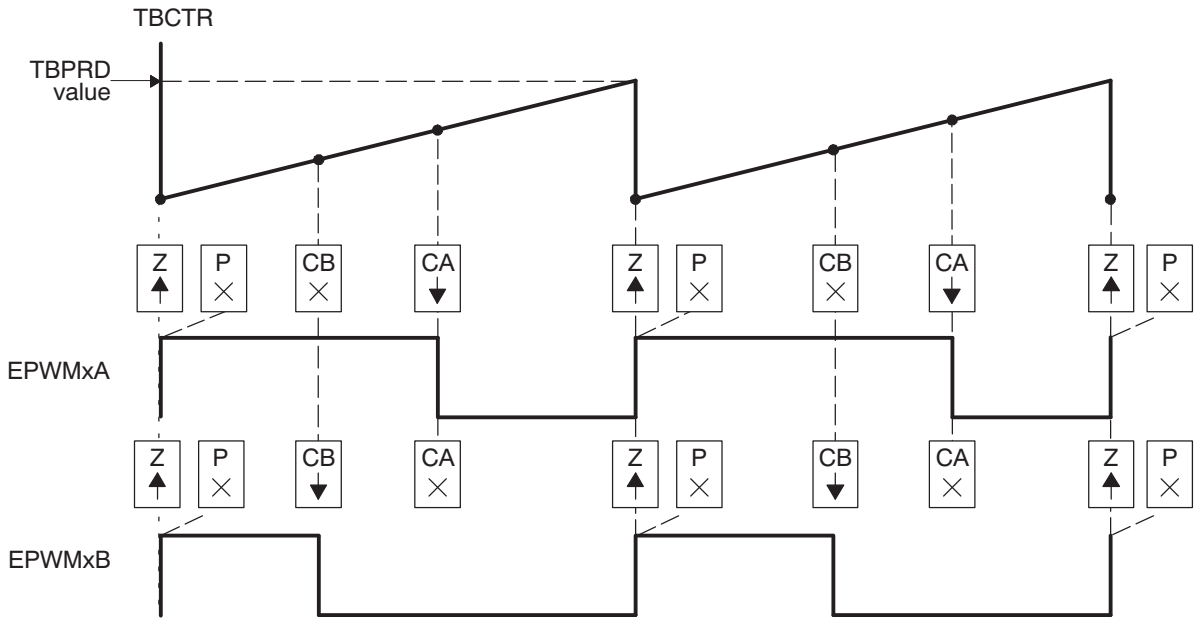


The PWM waveforms in Figure 20-20 through Figure 20-25 show some common action-qualifier configurations. The C-code samples in Example 20-1 through Example 20-6 shows how to configure an ePWM module for each case. Some conventions used in the figures and examples are as follows:

- TBPRD, CMPA, and CMPB refer to the value written in their respective registers. The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either CMPA or CMPB.
- EPWMxA and EPWMxB refer to the output signals from ePWMx
- Up-Down means Count-up-and-down mode, Up means up-count mode and Dwn means down-count mode
- Sym = Symmetric, Asym = Asymmetric

Example 20-1 contains a code sample showing initialization and run time for the waveforms in Figure 20-20.

Figure 20-20. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High



A PWM period = (TBPRD + 1) × T_{TBCLK}

B Duty modulation for EPWMxA is set by CMPA, and is active high (that is, high time duty proportional to CMPA).

C Duty modulation for EPWMxB is set by CMPB and is active high (that is, high time duty proportional to CMPB).

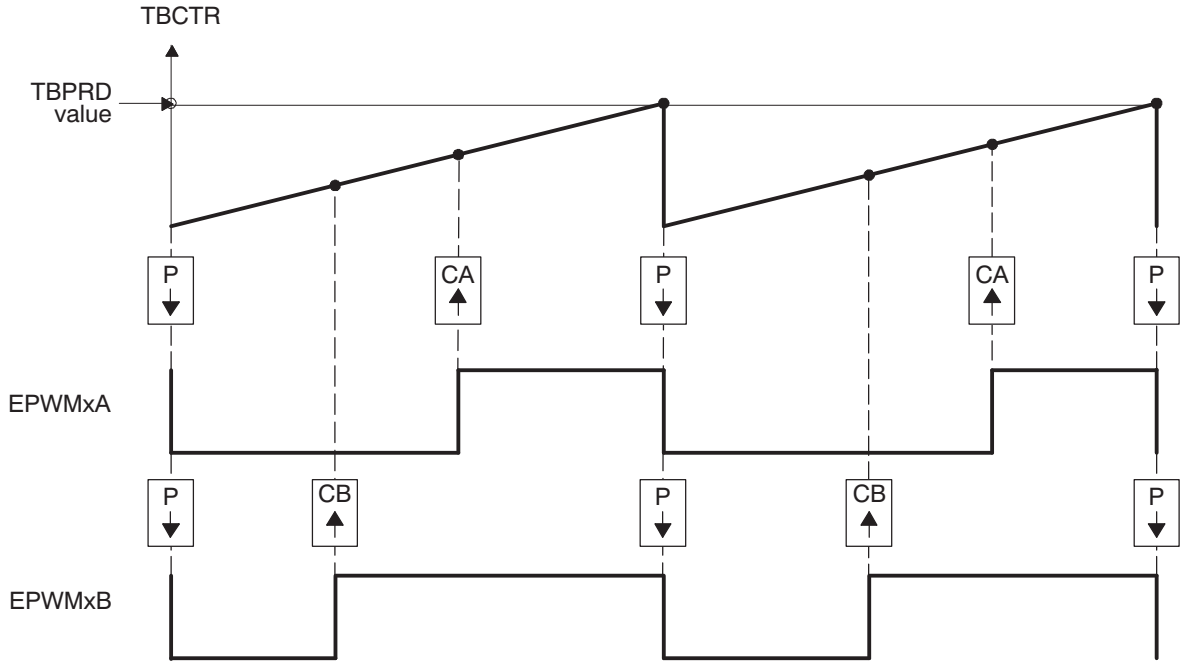
D The "Do Nothing" actions (X) are shown for completeness, but will not be shown on subsequent diagrams.

E Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Example 20-1. Code Sample for Figure 20-20

```
// Initialization Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.TBPRD = 600;                        // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350;               // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 200;                         // Compare B = 200 TBCLK counts
EPwm1Regs.TBPHS = 0;                          // Set Phase register to zero
EPwm1Regs.TBCTR = 0;                          // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;       // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;      // TBCLK = SYSCLK
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;
//
// Run Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.CMPA.half.CMPA = Duty1A;            // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B;                      // adjust duty for output EPWM1B
```

Figure 20-21. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low



A PWM period = (TBPRD + 1) × T_{TBCLK}

B Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

C Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

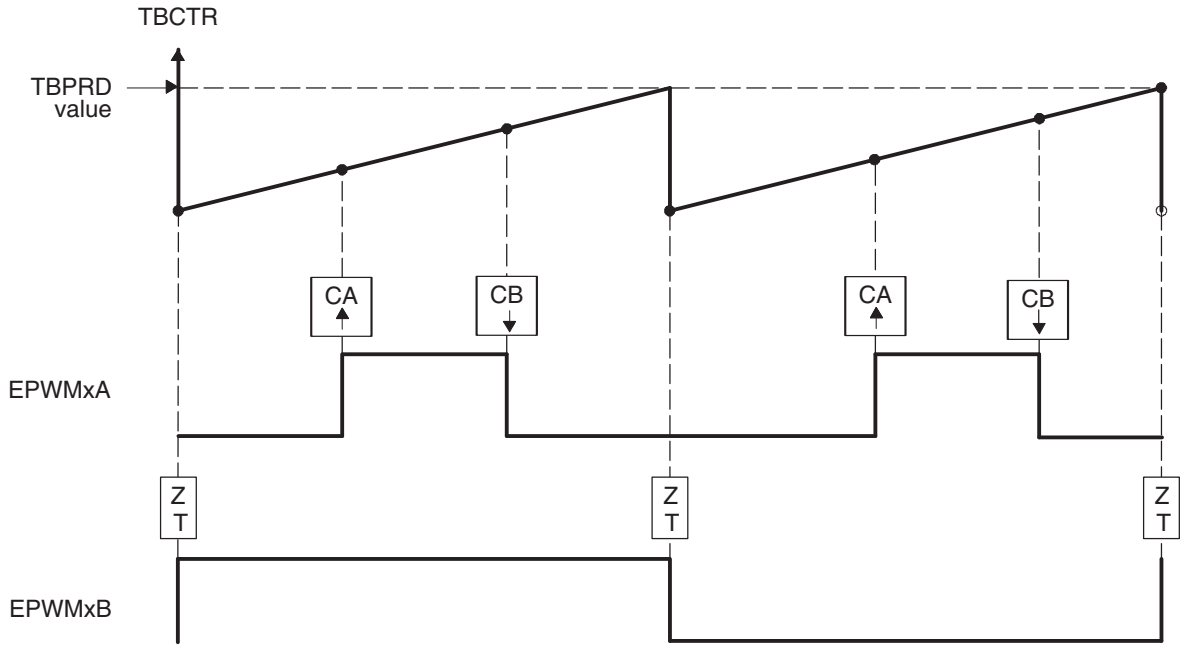
D Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Example 20-2 contains a code sample showing initialization and run time for the waveforms in Figure 20-21.

Example 20-2. Code Sample for Figure 20-21

```
// Initialization Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.TBPRD = 600;                          // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350;                 // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 200;                           // Compare B = 200 TBCLK counts
EPwm1Regs.TBPHS = 0;                            // Set Phase register to zero
EPwm1Regs.TBCTR = 0;                            // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;         // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;        // TBCLK = VCLK3
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.AQCTLA.bit.PRD = AQ_CLEAR;
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLB.bit.PRD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;
//
// Run Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.CMPA.half.CMPA = Duty1A;              // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B;                        // adjust duty for output EPWM1B
```

Figure 20-22. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA



A PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})

B Pulse can be placed anywhere within the PWM cycle (0000 - TBPRD)

C High time duty proportional to (CMPB - CMPA)

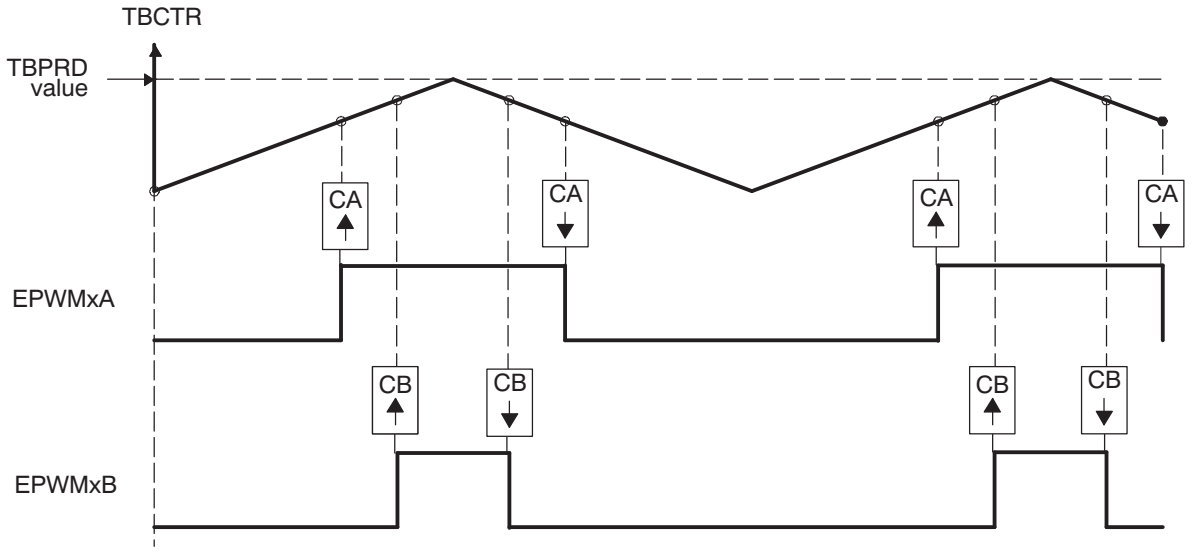
D EPWMxB can be used to generate a 50% duty square wave with frequency = 1/2 × ((TBPRD + 1) × TBCLK)

Example 20-3 contains a code sample showing initialization and run time for the waveforms Figure 20-22.

Example 20-3. Code Sample for Figure 20-22

```
// Initialization Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.TBPRD = 600;                          // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 200;                 // Compare A = 200 TBCLK counts
EPwm1Regs.CMPB = 400;                           // Compare B = 400 TBCLK counts
EPwm1Regs.TBPHS = 0;                            // Set Phase register to zero
EPwm1Regs.TBCTR = 0;                            // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;         // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;        // TBCLK = VCLK3
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CBU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_TOGGLE;
//
// Run Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.CMPA.half.CMPA = EdgePosA;          // adjust duty for output EPWM1A only
EPwm1Regs.CMPB = EdgePosB;
```

Figure 20-23. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low



A PWM period = 2 x TBPRD × T_{TBCLK}

B Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

C Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

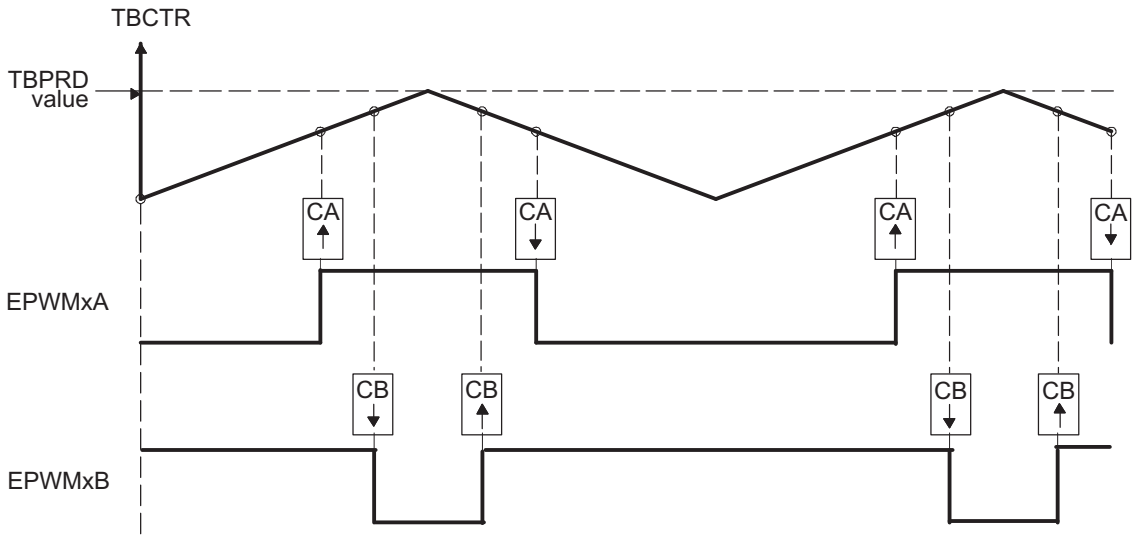
D Outputs EPWMxA and EPWMxB can drive independent power switches

Example 20-4 contains a code sample showing initialization and run time for the waveforms in Figure 20-23.

Example 20-4. Code Sample for Figure 20-23

```
// Initialization Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.TBPRD = 600;                          // Period = 2´600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 400;                 // Compare A = 400 TBCLK counts
EPwm1Regs.CMPB = 500;                           // Compare B = 500 TBCLK counts
EPwm1Regs.TBPHS = 0;                            // Set Phase register to zero
EPwm1Regs.TBCTR = 0;                            // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetric
xEPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;       // Phase loading disabled
xEPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;       // TBCLK = VCLK3
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;  // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;  // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;
EPwm1Regs.AQCTLB.bit.CBD = AQ_CLEAR;
//
// Run Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.CMPA.half.CMPA = Duty1A;            // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B;                      // adjust duty for output EPWM1B
```

Figure 20-24. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary



A PWM period = 2 × TBPRD × T_{TBCLK}

B Duty modulation for EPWMxA is set by CMPA, and is active low, i.e., low time duty proportional to CMPA

C Duty modulation for EPWMxB is set by CMPB and is active high, i.e., high time duty proportional to CMPB

D Outputs EPWMx can drive upper/lower (complementary) power switches

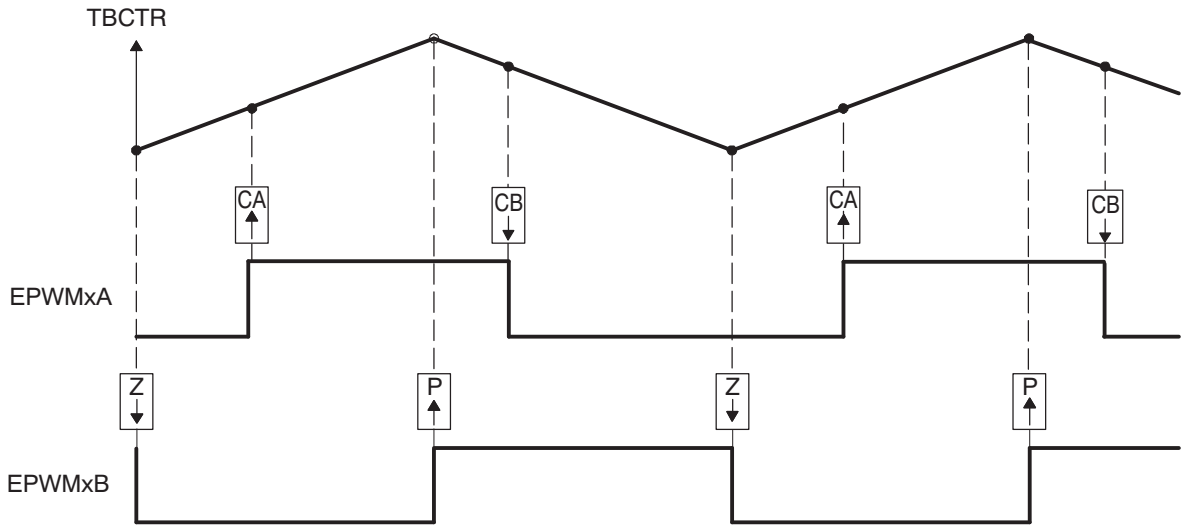
E Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

Example 20-5 contains a code sample showing initialization and run time for the waveforms in Figure 20-24.

Example 20-5. Code Sample for Figure 20-24

```
// Initialization Time
// = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.TBPRD = 600;                          // Period = 2´600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350;                 // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 400;                           // Compare B = 400 TBCLK counts
EPwm1Regs.TBPHS = 0;                            // Set Phase register to zero
EPwm1Regs.TBCTR = 0;                            // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetric
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;       // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;      // TBCLK = VCLK3
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBD = AQ_SET;
// Run Time
// = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.CMPA.half.CMPA = Duty1A;          // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B;                    // adjust duty for output EPWM1B
```

Figure 20-25. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low



A PWM period = 2 × TBPRD × TBCLK

B Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.

C Duty modulation for EPWMxA is set by CMPA and CMPB.

D Low time duty for EPWMxA is proportional to (CMPA + CMPB).

E To change this example to active high, CMPA and CMPB actions need to be inverted (i.e., Set ! Clear and Clear Set).

F Duty modulation for EPWMxB is fixed at 50% (utilizes spare action resources for EPWMxB)

Example 20-6 contains a code sample showing initialization and run time for the waveforms in Figure 20-25.

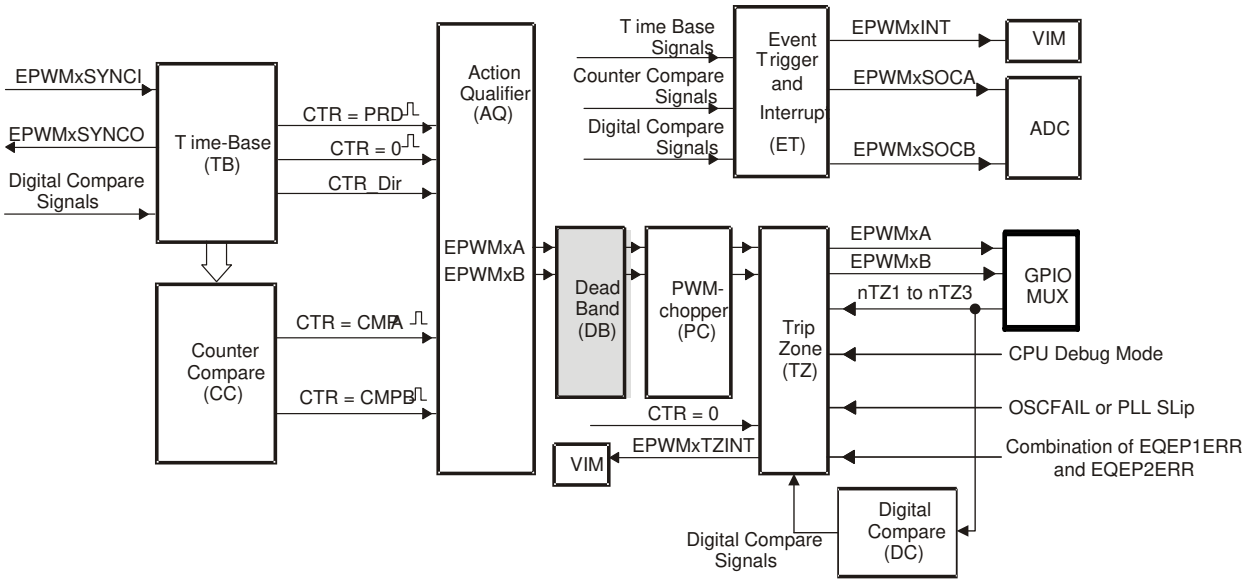
Example 20-6. Code Sample for Figure 20-25

```
// Initialization Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.TBPRD = 600;                          //  Period = 2 ´ 600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 250;                 //  Compare A = 250 TBCLK counts
EPwm1Regs.CMPB = 450;                           //  Compare B = 450 TBCLK counts
EPwm1Regs.TBPHS = 0;                            //  Set Phase register to zero
EPwm1Regs.TBCTR = 0;                            //  clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; //  Symmetric
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;         //  Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;        //  TBCLK = VCLK3
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;  //  load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;  //  load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CBD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.PRD = AQ_SET;
// Run Time
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
EPwm1Regs.CMPA.half.CMPA = EdgePosA;            // adjust duty for output EPWM1A only
EPwm1Regs.CMPB = EdgePosB;
```

20.2.5 Dead-Band Generator (DB) Submodule

Figure 20-26 illustrates the dead-band submodule within the ePWM module.

Figure 20-26. Dead_Band Submodule



20.2.5.1 Purpose of the Dead-Band Submodule

The "Action-qualifier (AQ) Module" section discussed how it is possible to generate the required dead-band by having full control over edge placement using both the CMPA and CMPB resources of the ePWM module. However, if the more classical edge delay-based dead-band with polarity control is required, then the dead-band submodule described here should be used.

The key functions of the dead-band module are:

- Generating appropriate signal pairs (EPWMxA and EPWMxB) with dead-band relationship from a single EPWMxA input
- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

20.2.5.2 Controlling and Monitoring the Dead-Band Submodule

The dead-band submodule operation is controlled and monitored via the following registers:

Table 20-13. Dead-Band Generator Submodule Registers

Register Name	Address Offset	Shadowed	Description
DBCTL		No	Dead-Band Control Register
DBFED		No	Dead-Band Falling Edge Delay Count Register
DBRED		No	Dead-Band Rising Edge Delay Count Register

20.2.5.3 Operational Highlights for the Dead-Band Submodule

The following sections provide the operational highlights.

The dead-band submodule has two groups of independent selection options as shown in Figure 20-27.

- **Input Source Selection:**

  The input signals to the dead-band module are the EPWMxA and EPWMxB output signals from the action-qualifier. In this section they will be referred to as EPWMxA In and EPWMxB In. Using the DBCTL[IN_MODE) control bits, the signal source for each delay, falling-edge or rising-edge, can be selected:

  – EPWMxA In is the source for both falling-edge and rising-edge delay. This is the default mode.
  – EPWMxA In is the source for falling-edge delay, EPWMxB In is the source for rising-edge delay.
  – EPWMxA In is the source for rising edge delay, EPWMxB In is the source for falling-edge delay.
  – EPWMxB In is the source for both falling-edge and rising-edge delay.

- **Half Cycle Clocking:**

  The dead-band submodule can be clocked using half cycle clocking to double the resolution (i.e. counter clocked at 2× TBCLK)

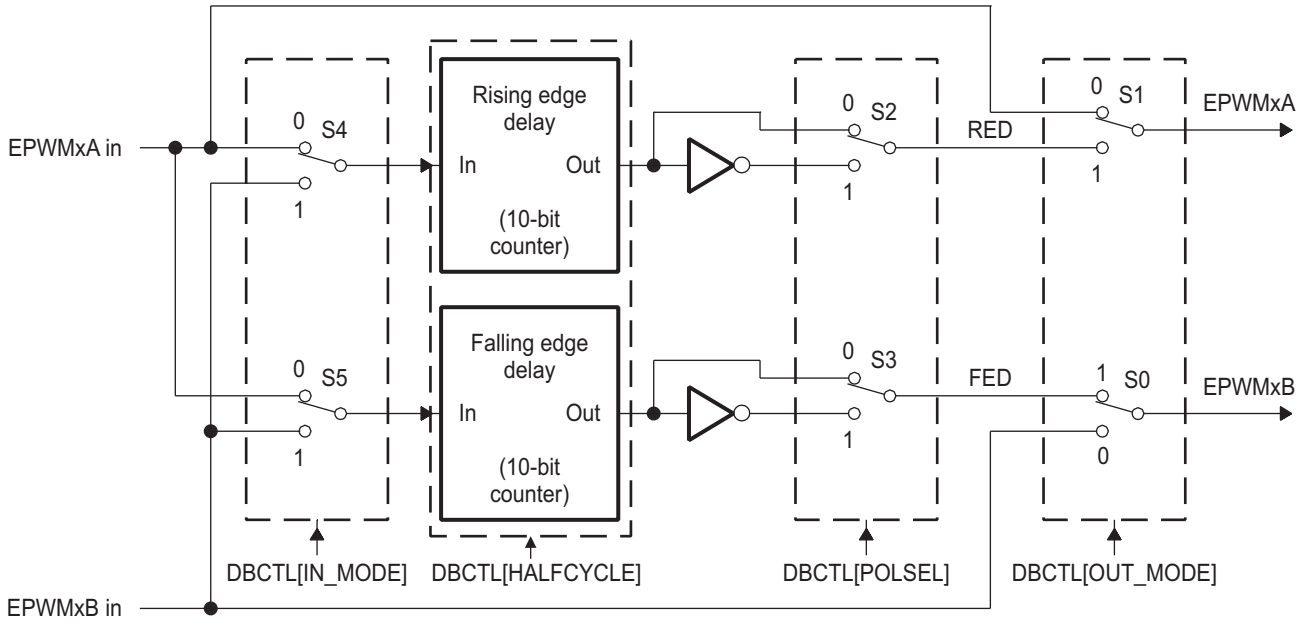
- **Output Mode Control:**

  The output mode is configured by way of the DBCTL[OUT_MODE] bits. These bits determine if the falling-edge delay, rising-edge delay, neither, or both are applied to the input signals.

- **Polarity Control:**

  The polarity control (DBCTL[POLSEL]) allows you to specify whether the rising-edge delayed signal and/or the falling-edge delayed signal is to be inverted before being sent out of the dead-band submodule.

Figure 20-27. Configuration Options for the Dead-Band Submodule



Although all combinations are supported, not all are typical usage modes. Table 20-14 documents some classical dead-band configurations. These modes assume that the DBCTL[IN_MODE] is configured such that EPWMxA In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 20-14 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED)**

  Allows you to fully disable the dead-band submodule from the PWM signal path.
- **Mode 2-5: Classical Dead-Band Polarity Settings:**

  These represent typical polarity configurations that should address all the active high/low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 20-28. Note that to generate equivalent waveforms to Figure 20-28, configure the action-qualifier submodule to generate the signal as shown for EPWMxA.
- **Mode 6: Bypass rising-edge-delay and Mode 7: Bypass falling-edge-delay**

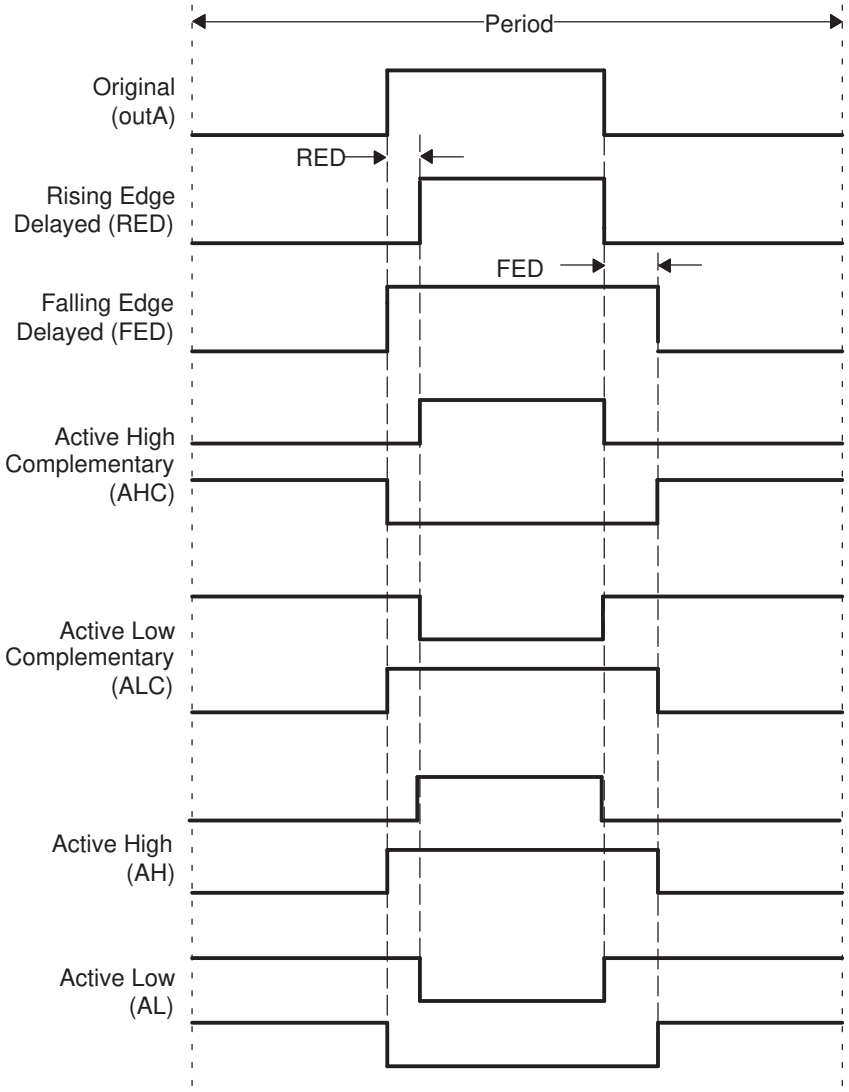
  Finally the last two entries in Table 20-14 show combinations where either the falling-edge-delay (FED) or rising-edge-delay (RED) blocks are bypassed.

Table 20-14. Classical Dead-Band Operating Modes

		DBCTL[POLSEL]		DBCTL[OUT_MODE]	
Mode	**Mode Description**	**S3**	**S2**	**S1**	**S0**
1	EPWMxA and EPWMxB Passed Through (No Delay)	X	X	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	EPWMxA Out = EPWMxA In (No Delay) EPWMxB Out = EPWMxA In with Falling Edge Delay	0 or 1	0 or 1	0	1
7	EPWMxA Out = EPWMxA In with Rising Edge Delay EPWMxB Out = EPWMxB In with No Delay	0 or 1	0 or 1	1	0

Figure 20-28 shows waveforms for typical cases where 0% < duty < 100%.

Figure 20-28. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)



The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the DBRED and DBFED registers. These are 10-bit registers and their value represents the number of time-base clock, TBCLK, periods a signal edge is delayed by. For example, the formula to calculate falling-edge-delay and rising-edge-delay are:

$FED = DBFED \times T_{TBCLK}$

$RED = DBRED \times T_{TBCLK}$

Where T_{TBCLK} is the period of TBCLK, the prescaled version of VCLK3.

For convenience, delay values for various TBCLK options are shown in Table 20-15.

Table 20-15. Dead-Band Delay Values in µS as a Function of DBFED and DBRED

Dead-Band Value	Dead-Band Delay in µS		
DBFED, DBRED	TBCLK = VCLK3/1	TBCLK = VCLK3 /2	TBCLK = VCLK3/4
1	0.02 µS	0.03 µS	0.07 µS
5	0.08 µS	0.17 µS	0.33 µS
10	0.17 µS	0.33 µS	0.67 µS
100	1.67 µS	3.33 µS	6.67 µS
200	3.33 µS	6.67 µS	13.33 µS
400	6.67 µS	13.33 µS	26.67 µS
500	8.33 µS	16.67 µS	33.33 µS
600	10.00 µS	20.00 µS	40.00 µS
700	11.67 µS	23.33 µS	46.67 µS
800	13.33 µS	26.67 µS	53.33 µS
900	15.00 µS	30.00 µS	60.00 µS
1000	16.67 µS	33.33 µS	66.67 µS

When half-cycle clocking is enabled, the formula to calculate the falling-edge-delay and rising-edge-delay becomes:

$FED = DBFED \times T_{TBCLK}/2$

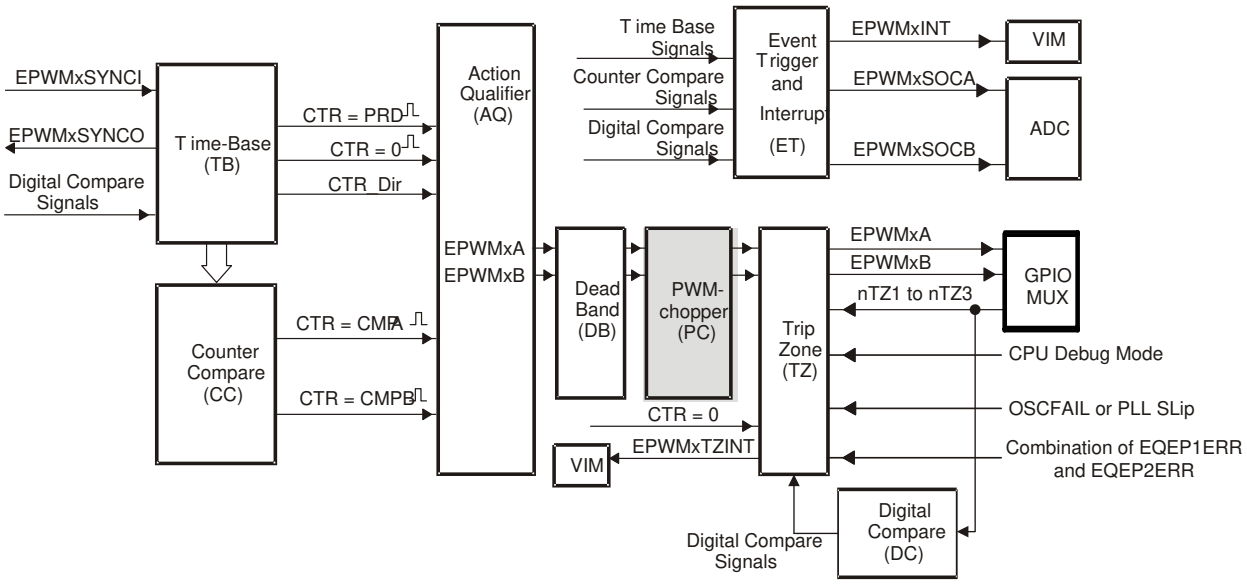
$RED = DBRED \times T_{TBCLK}/2$

20.2.6 PWM-Chopper (PC) Submodule

Figure 20-29 illustrates the PWM-chopper (PC) submodule within the ePWM module.

The PWM-chopper submodule allows a high-frequency carrier signal to modulate the PWM waveform generated by the action-qualifier and dead-band submodules. This capability is important if you need pulse transformer-based gate drivers to control the power switching elements.

Figure 20-29. PWM-Chopper Submodule



20.2.6.1 Purpose of the PWM-Chopper Submodule

The key functions of the PWM-chopper submodule are:

- Programmable chopping (carrier) frequency
- Programmable pulse width of first pulse
- Programmable duty cycle of second and subsequent pulses
- Can be fully bypassed if not required

20.2.6.2 Controlling the PWM-Chopper Submodule

The PWM-chopper submodule operation is controlled via the registers in Table 20-16.

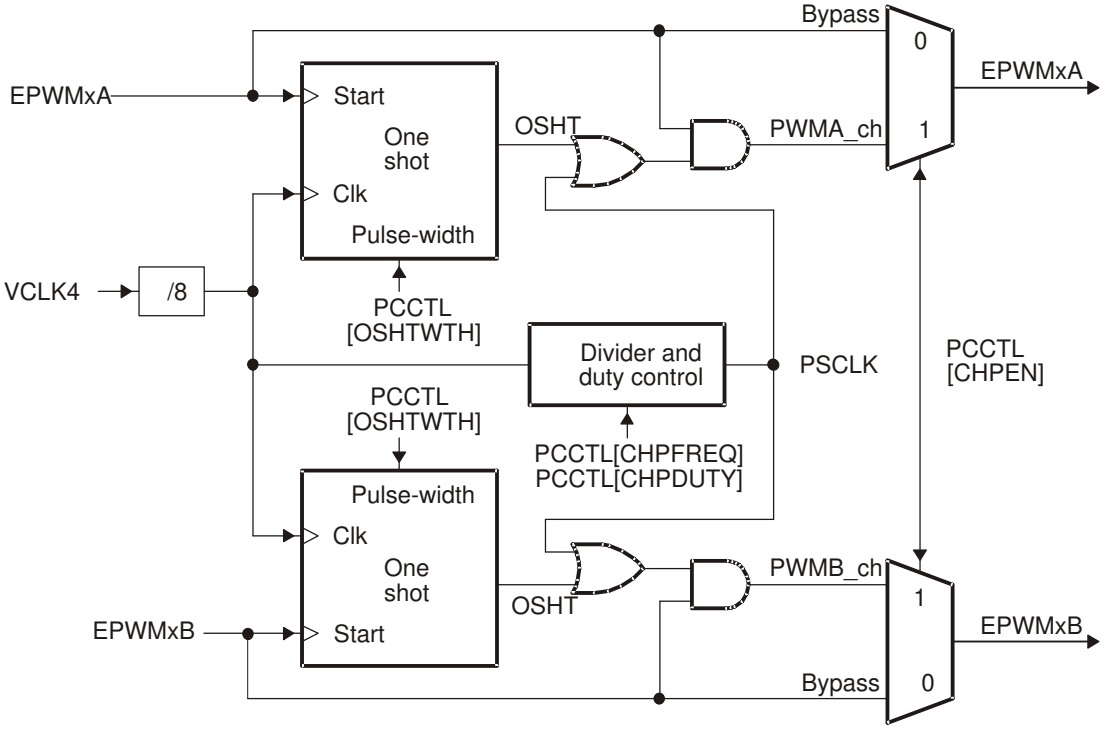
Table 20-16. PWM-Chopper Submodule Registers

Register Name	Address Offset	Shadowed	Description
PCCTL		No	PWM-chopper Control Register

20.2.6.3 Operational Highlights for the PWM-Chopper Submodule

Figure 20-30 shows the operational details of the PWM-chopper submodule. The carrier clock is derived from VCLK3. Its frequency and duty cycle are controlled via the CHPFREQ and CHPDUTY bits in the PCCTL register. The one-shot block is a feature that provides a high energy first pulse to ensure hard and fast power switch turn on, while the subsequent pulses sustain pulses, ensuring the power switch remains on. The one-shot width is programmed via the OSHTWTH bits. The PWM-chopper submodule can be fully disabled (bypassed) via the CHPEN bit.

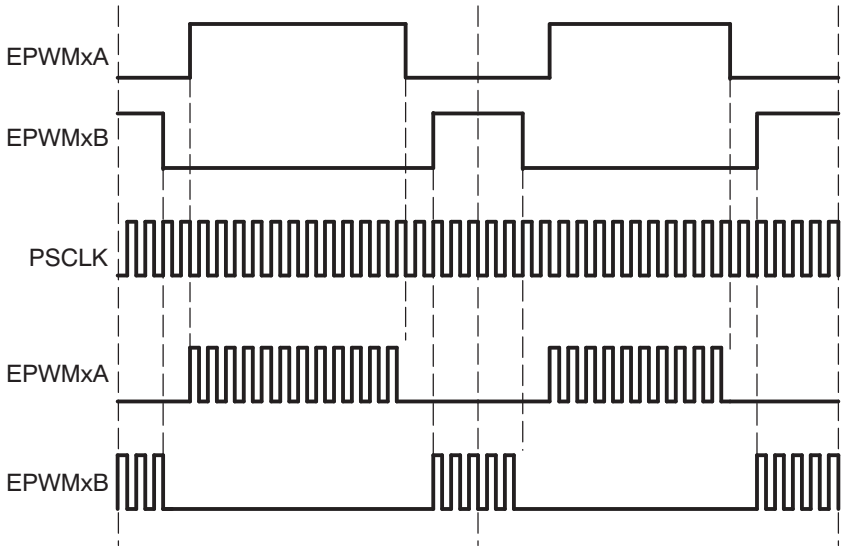
Figure 20-30. PWM-Chopper Submodule Operational Details



20.2.6.4 Waveforms

Figure 20-31 shows simplified waveforms of the chopping action only; one-shot and duty-cycle control are not shown. Details of the one-shot and duty-cycle control are discussed in the following sections.

Figure 20-31. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only



20.2.6.4.1 One-Shot Pulse

The width of the first pulse can be programmed to any of 16 possible pulse width values. The width or period of the first pulse is given by:

$T_{1stpulse} = T_{VCLK3} \times 8 \times OSHTWTH$

Where T_{VCLK3} is the period of the system clock (VCLK3) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 20-32 shows the first and subsequent sustaining pulses and Table 20-17 gives the possible pulse width values for a VCLK3 = 100 MHz.

Figure 20-32. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses

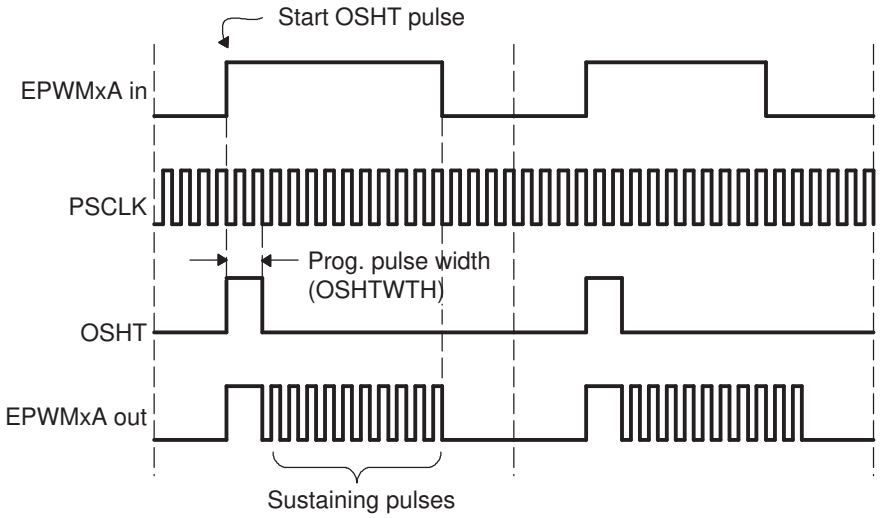


Table 20-17. Possible Pulse Width Values for VCLK3 = 100 MHz

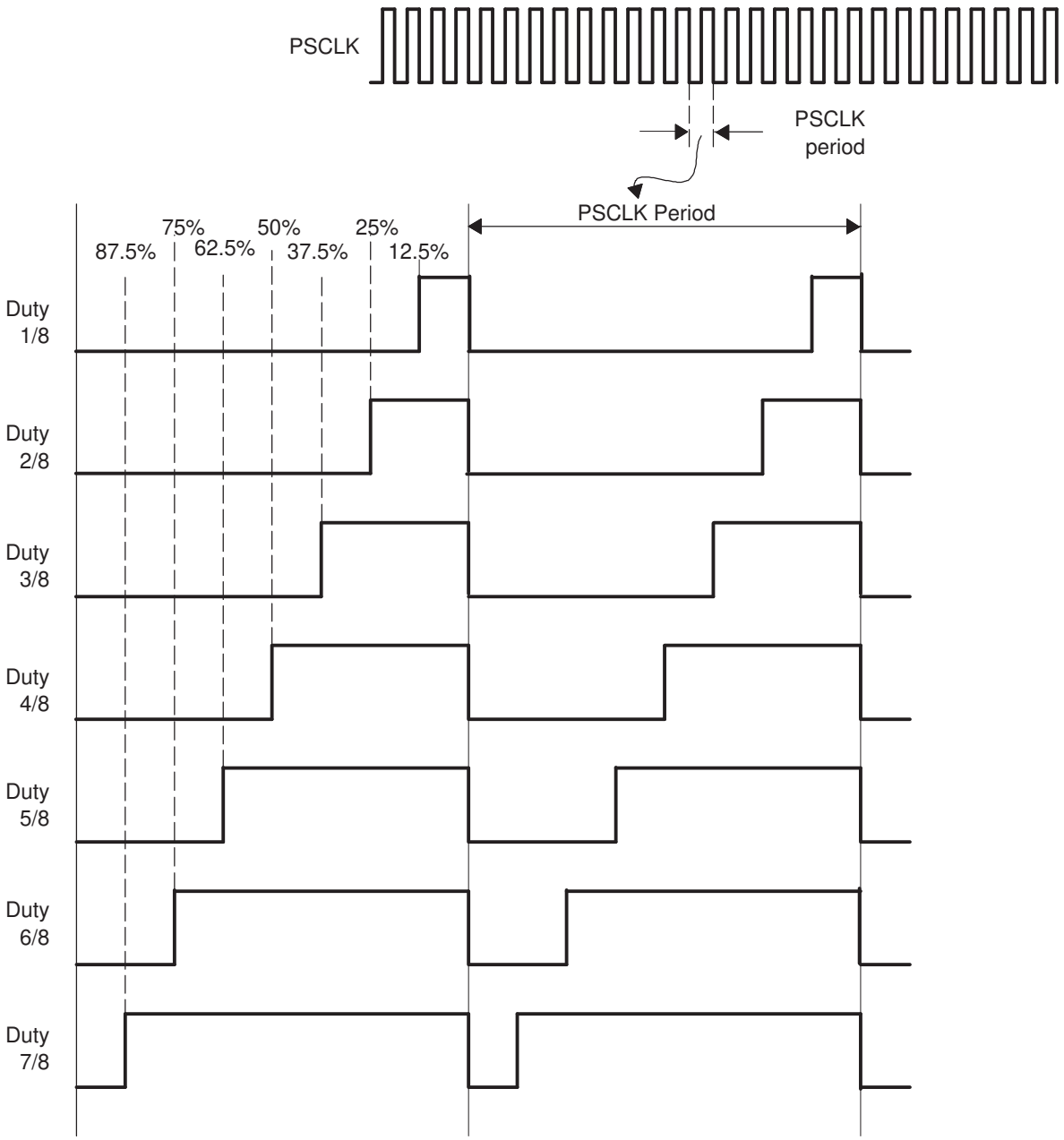
OSHTWTHz (hex)	Pulse Width (nS)
0	100
1	200
2	300
3	400
4	500
5	600
6	700
7	800
8	900
9	1000
A	1100
B	1200
C	1300
D	1400
E	1500
F	1600

20.2.6.4.2 Duty Cycle Control

Pulse transformer-based gate drive designs need to comprehend the magnetic properties or characteristics of the transformer and associated circuitry. Saturation is one such consideration. To assist the gate drive designer, the duty cycles of the second and subsequent pulses have been made programmable. These sustaining pulses ensure the correct drive strength and polarity is maintained on the power switch gate during the on period, and hence a programmable duty cycle allows a design to be tuned or optimized via software control.

Figure 20-33 shows the duty cycle control that is possible by programming the CHPDUTY bits. One of seven possible duty ratios can be selected ranging from 12.5% to 87.5%.

Figure 20-33. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses

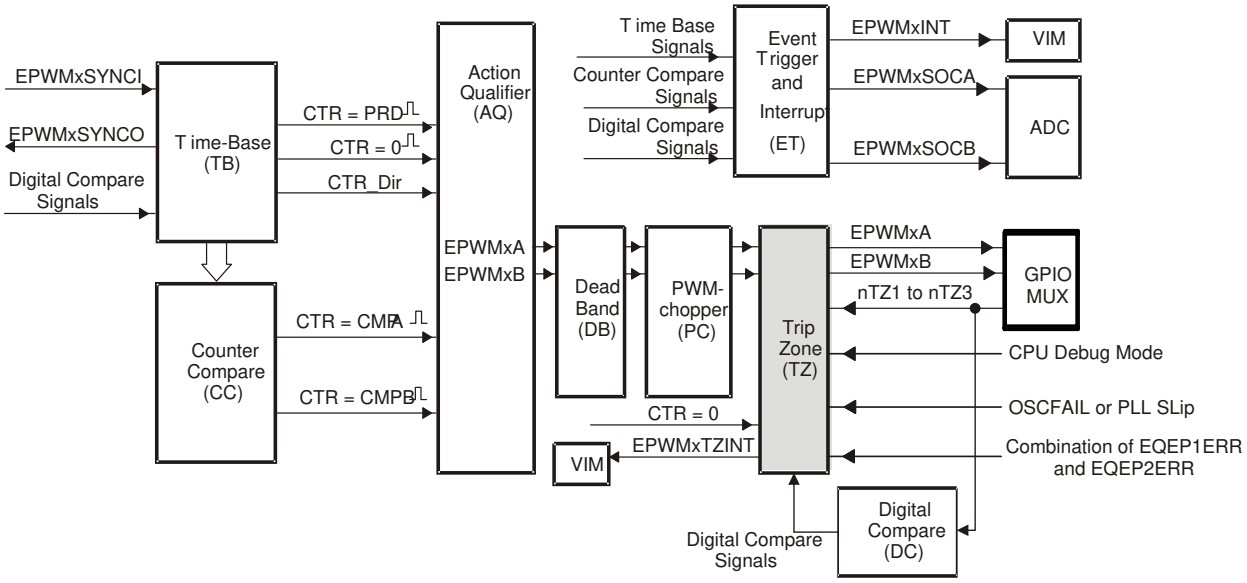


20.2.7 Trip-Zone (TZ) Submodule

Figure 20-34 shows how the trip-zone (TZ) submodule fits within the ePWM module.

Each ePWM module is connected to six TZn signals (TZ1 to TZ6). TZ1 to TZ3 are sourced from the GPIO mux. TZ4 is sourced from a combination of EQEP1ERR and EQEP2ERR signals. TZ5 is connected to the system oscillator or PLL clock fail logic, and TZ6 is sourced from the debug mode halt indication output from the CPU. These signals indicate fault or trip conditions, and the ePWM outputs can be programmed to respond accordingly when faults occur.

Figure 20-34. Trip-Zone Submodule



20.2.7.1 Purpose of the Trip-Zone Submodule

The key functions of the Trip-Zone submodule are:

- Trip inputs TZ1 to TZ6 are mapped to all ePWM modules.
- Upon a fault indication, either no action is taken or the ePWM outputs EPWMxA and EPWMxB can be forced to one of the following:
  - High
  - Low
  - High-impedance
- Support for one-shot trip (OSHT) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Support for digital compare tripping (DC) based on state of on-chip analog comparator module outputs and/or TZ1 to TZ3 signals.
- Each trip-zone input and digital compare (DC) submodule DCAEVT1/2 or DCBEVT1/2 force event can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone input.
- Software-forced tripping is also supported.
- The trip-zone submodule can be fully bypassed if it is not required.

20.2.7.2 Controlling and Monitoring the Trip-Zone Submodule

The trip-zone submodule operation is controlled and monitored through the following registers:

Table 20-18. Trip-Zone Submodule Registers

Register Name	Address Offset	Shadowed	Description [1]
TZDCSEL		No	Trip-zone Digital Compare Select Register [2]
TZSEL		No	Trip-Zone Select Register
TZEINT		No	Trip-Zone Enable Interrupt Register
TZCTL		No	Trip-Zone Control Register
TZCLR		No	Trip-Zone Clear Register
TZFLG		No	Trip-Zone Flag Register
TZFRC		No	Trip-Zone Force Register

(1) All trip-zone registers are writable only in privileged mode.
(2) This register is discussed in more detail in Section 20.2.9.

20.2.7.3 Operational Highlights for the Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals $\overline{TZ1}$ to $\overline{TZ6}$ (also collectively referred to as $\overline{TZn}$) are active low input signals. When one of these signals goes low, or when a DCAEVT1/2 or DCBEVT1/2 force happens based on the TZDCSEL register event selection, it indicates that a trip event has occurred. Each ePWM module can be individually configured to ignore or use each of the trip-zone signals or DC events. Which trip-zone signals or DC events are used by a particular ePWM module is determined by the TZSEL register for that specific ePWM module. The trip-zone signals may or may not be synchronized to the system clock (VCLK3) and digitally filtered within the GPIO MUX block. A minimum of 3 × TBCLK low pulse width on $\overline{TZn}$ inputs is sufficient to trigger a fault condition on the ePWM module. If the pulse width is less than this, the trip condition may not be latched. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on $\overline{TZn}$ inputs. The GPIOs or peripherals must be appropriately configured. For more information, see the IOMM chapter of the device technical reference manual.

Each $\overline{TZn}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for an ePWM module. DCAEVT1 and DCBEVT1 events can be configured to directly trip an ePWM module or provide a one-shot trip event to the module. Likewise, DCAVET2 and DCBEVT2 events can also be configured to directly trip an ePWM module or provide a cycle-by-cycle trip event to the module. This configuration is determined by the TZSEL[DCAEVT1/2], TZSEL[DCBEVT1/2], TZSEL[CBCn], and TZSEL[OSHTn] control bits (where n corresponds to the trip input) respectively.

- **Cycle-by-Cycle (CBC):**

  When a cycle-by-cycle trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the EPWMxA and/or EPWMxB output. Table 20-19 lists the possible actions. In addition, the cycle-by-cycle trip event flag (TZFLG[CBC]) is set and a EPWMx_TZINT interrupt is generated if it is enabled in the TZEINT register and VIM peripheral.

  If the CBC interrupt is enabled via the TZEINT register, and DCAEVT2 or DCBEVT2 are selected as CBC trip sources via the TZSEL register, it is not necessary to also enable the DCAEVT2 or DCBEVT2 interrupts in the TZEINT register, as the DC events trigger interrupts through the CBC mechanism.

  The specified condition on the inputs is automatically cleared when the ePWM time-base counter reaches zero (TBCTR = 0x0000) if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The TZFLG[CBC] flag bit will remain set until it is manually cleared by writing to the TZCLR[CBC] bit. If the cycle-by-cycle trip event is still present when the TZFLG[CBC] bit is cleared, then it will again be immediately set.

- **One-Shot (OSHT):**

  When a one-shot trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the EPWMxA and/or EPWMxB output. Table 20-19 lists the possible actions. In addition, the one-shot trip event flag (TZFLG[OST]) is set and a EPWMx_TZINT interrupt is generated if it is enabled in the TZEINT register and VIM peripheral. The one-shot trip condition must be cleared manually by writing to the TZCLR[OST] bit.

  If the one-shot interrupt is enabled via the TZEINT register, and DCAEVT1 or DCBEVT1 are selected as OSHT trip sources via the TZSEL register, it is not necessary to also enable the DCAEVT1 or DCBEVT1 interrupts in the TZEINT register, as the DC events trigger interrupts through the OSHT mechanism.

- **Digital Compare Events (DCAEVT1/2 and DCBEVT1/2):**

  A digital compare DCAEVT1/2 or DCBEVT1/2 event is generated based on a combination of the DCAH/DCAL and DCBH/DCBL signals as selected by the TZDCSEL register. The signals which source the DCAH/DCAL and DCBH/DCBL signals are selected via the DCTRIPSEL register and can be either trip zone input pins. For more information on the digital compare submodule signals, see Section 20.2.9.

  When a digital compare event occurs, the action specified in the TZCTL[DCAEVT1/2] and TZCTL[DCBEVT1/2] bits is carried out immediately on the EPWMxA and/or EPWMxB output. Table 20-19 lists the possible actions. In addition, the relevant DC trip event flag (TZFLG[DCAEVT1/2] / TZFLG[DCBEVT1/2]) is set and a EPWMx_TZINT interrupt is generated if it is enabled in the TZEINT register and VIM peripheral.

  The specified condition on the pins is automatically cleared when the DC trip event is no longer present. The TZFLG[DCAEVT1/2] or TZFLG[DCBEVT1/2] flag bit will remain set until it is manually cleared by writing to the TZCLR[DCAEVT1/2] or TZCLR[DCBEVT1/2] bit. If the DC trip event is still present when the TZFLG[DCAEVT1/2] or TZFLG[DCBEVT1/2] flag is cleared, then it will again be immediately set.

The action taken when a trip event occurs can be configured individually for each of the ePWM output pins by way of the TZCTL register bit fields. One of four possible actions, shown in Table 20-19, can be taken on a trip event.

Table 20-19. Possible Actions On a Trip Event

TZCTL Register bit-field Settings	EPWMxA and/or EPWMxB	Comment
0,0	High-Impedance	Tripped
0,1	Force to High State	Tripped
1,0	Force to Low State	Tripped
1,1	No Change	Do Nothing. No change is made to the output.

Example 20-7. Trip-Zone Configurations

Scenario A:
A one-shot trip event on $\overline{TZ1}$ pulls both EPWM1A, EPWM1B low and also forces EPWM2A and EPWM2B high.

- Configure the ePWM1 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ1}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A will be forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B will be forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ1}$ as a one-shot event source for ePWM2
  - TZCTL[TZA] = 1: EPWM2A will be forced high on a trip event.
  - TZCTL[TZB] = 1: EPWM2B will be forced high on a trip event.

Scenario B:
A cycle-by-cycle event on $\overline{TZ5}$ pulls both EPWM1A, EPWM1B low.
A one-shot event on $\overline{TZ1}$ or $\overline{TZ6}$ puts EPWM2A into a high impedance state.

- Configure the ePWM1 registers as follows:
  - TZSEL[CBC5] = 1: enables $\overline{TZ5}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A will be forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B will be forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ1}$ as a one-shot event source for ePWM2
  - TZSEL[OSHT6] = 1: enables $\overline{TZ6}$ as a one-shot event source for ePWM2
  - TZCTL[TZA] = 0: EPWM2A will be put into a high-impedance state on a trip event.
  - TZCTL[TZB] = 3: EPWM2B will ignore the trip event.

20.2.7.4 Generating Trip Event Interrupts

Figure 20-35 and Figure 20-36 illustrate the trip-zone submodule control and interrupt logic, respectively. DCAEVT1/2 and DCBEVT1/2 signals are described in further detail in Section 20.2.9.

Figure 20-35. Trip-Zone Submodule Mode Control Logic

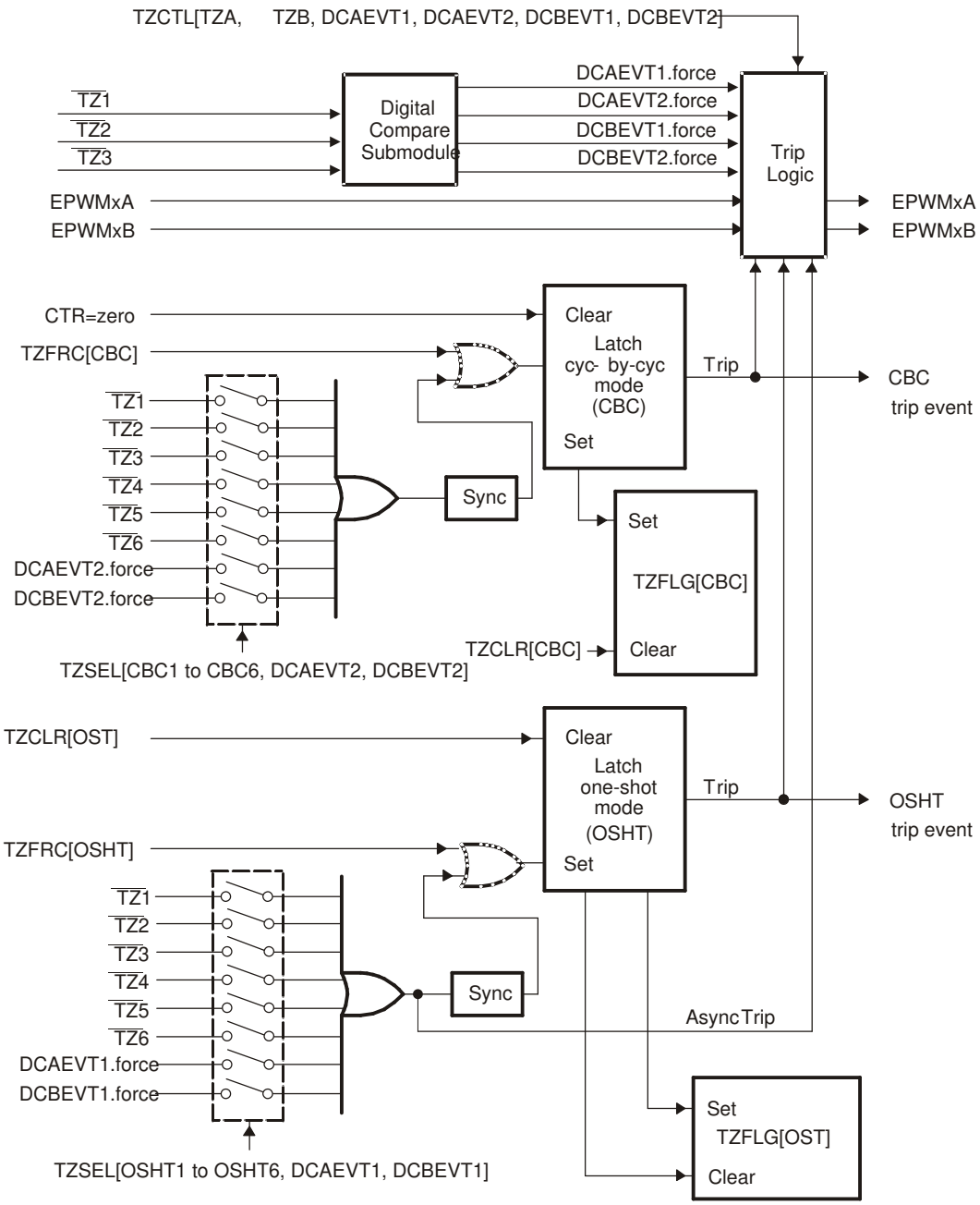
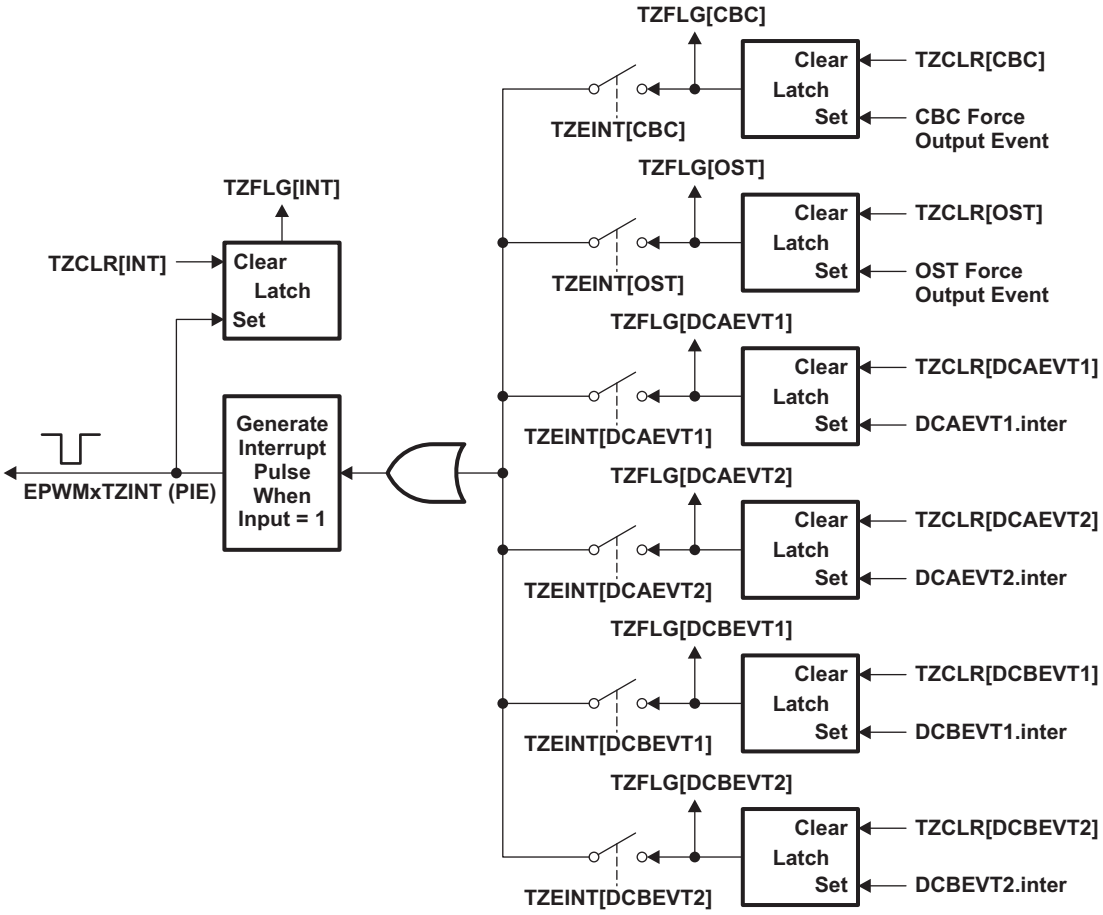


Figure 20-36. Trip-Zone Submodule Interrupt Logic



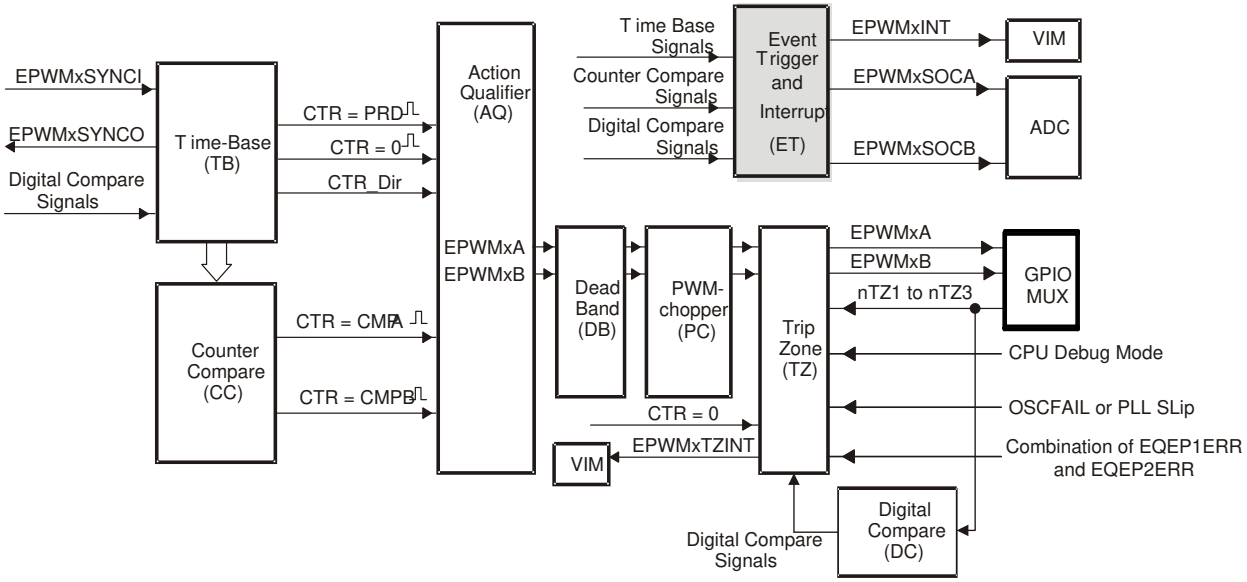
20.2.8 Event-Trigger (ET) Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base, counter-compare and digital-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests and ADC start of conversion at:
  - Every event
  - Every second event
  - Every third event
- Provides full visibility of event generation via event counters and flags
- Allows software forcing of Interrupts and ADC start of conversion

The event-trigger submodule manages the events generated by the time-base submodule, the counter-compare submodule, and the digital-compare submodule to generate an interrupt to the CPU and/or a start of conversion pulse to the ADC when a selected event occurs. Figure 20-37 illustrates where the event-trigger submodule fits within the ePWM system.

Figure 20-37. Event-Trigger Submodule

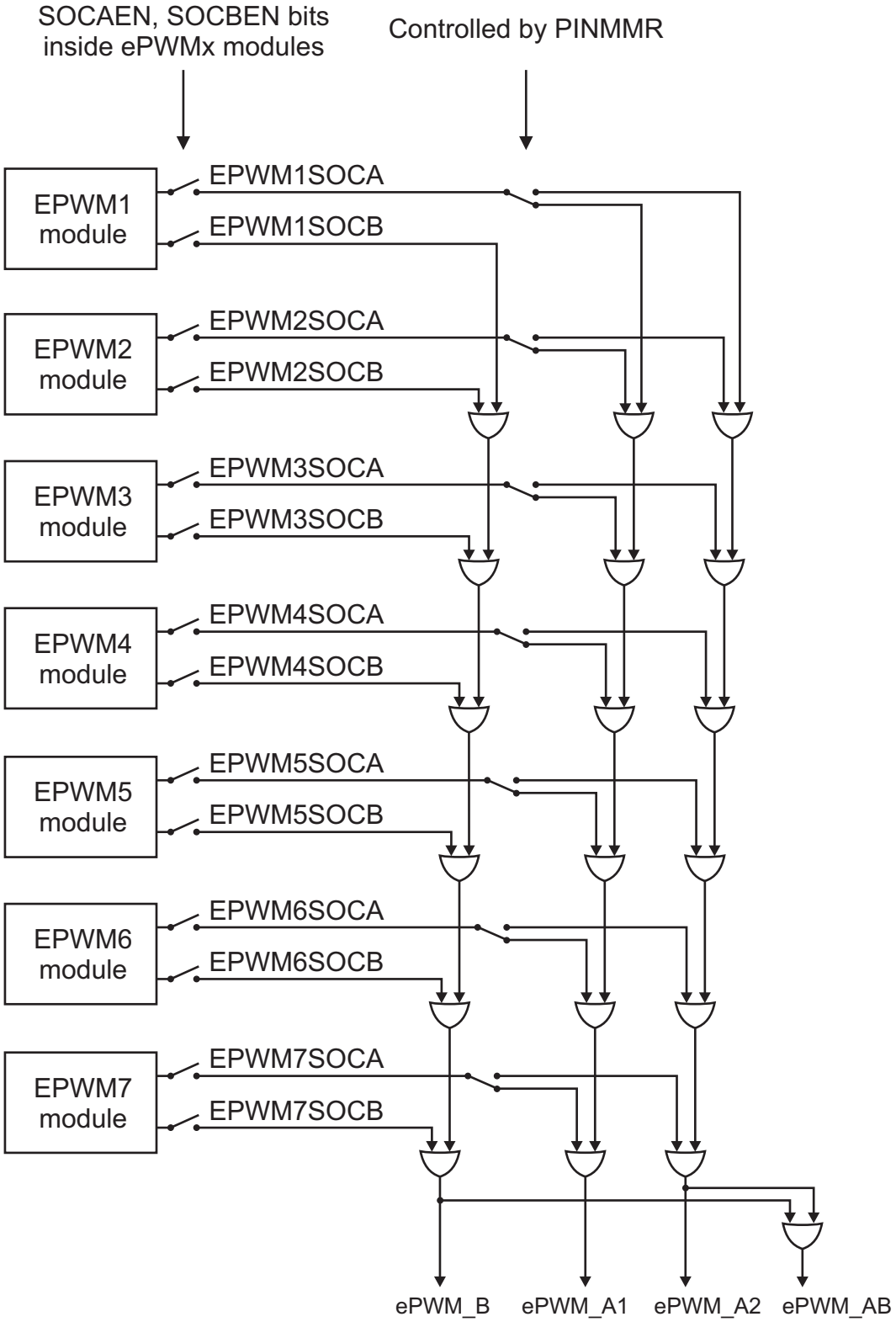


20.2.8.1 Operational Overview of the Event-Trigger Submodule

The following sections describe the event-trigger submodule's operational highlights.

Each ePWM module has one interrupt request line connected to the VIM and two start of conversion signals connected to the ADC module. As shown in Figure 20-38, the ePWMxSOCA and ePWMxSOCB signals are combined to generate four special signals that can be used to trigger an ADC start of conversion, and hence multiple modules can initiate an ADC start of conversion via the ADC trigger inputs.

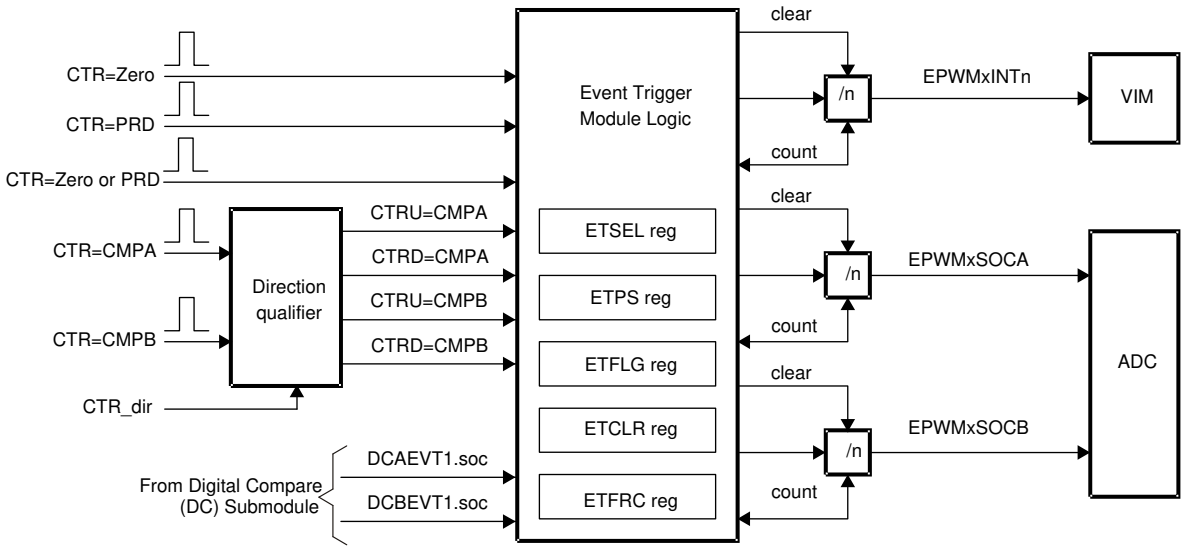
Figure 20-38. Event-Trigger Submodule Inter-Connectivity of ADC Start of Conversion



The event-trigger submodule monitors various event conditions (the left side inputs to event-trigger submodule shown in Figure 20-39) and can be configured to prescale these events before issuing an Interrupt request or an ADC start of conversion. The event-trigger prescaling logic can issue Interrupt requests and ADC start of conversion at:

- Every event
- Every second event
- Every third event

Figure 20-39. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs



The key registers used to configure the event-trigger submodule are listed in Table 20-20.

Table 20-20. Event-Trigger Submodule Registers

Register Name	Address Offset	Shadowed	Description
ETSEL		No	Event-trigger Selection Register
ETFLG		No	Event-trigger Flag Register
ETPS		No	Event-trigger Prescale Register
ETFRC		No	Event-trigger Force Register
ETCLR		No	Event-trigger Clear Register

- ETSEL—This selects which of the possible events will trigger an interrupt or start an ADC conversion
- ETPS—This programs the event prescaling options mentioned above.
- ETFLG—These are flag bits indicating status of the selected and prescaled events.
- ETCLR—These bits allow you to clear the flag bits in the ETFLG register via software.
- ETFRC—These bits allow software forcing of an event. Useful for debugging or s/w intervention.

A more detailed look at how the various register bits interact with the Interrupt and ADC start of conversion logic are shown in Figure 20-40, Figure 20-41, and Figure 20-42.

Figure 20-40 shows the event-trigger's interrupt generation logic. The interrupt-period (ETPS[INTPRD]) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt.
- Generate an interrupt on every event
- Generate an interrupt on every second event
- Generate an interrupt on every third event

Which event can cause an interrupt is configured by the interrupt selection (ETSEL[INTSEL]) bits. The event can be one of the following:

- Time-base counter equal to zero (TBCTR = 0x0000).
- Time-base counter equal to period (TBCTR = TBPRD).
- Time-base counter equal to zero or period (TBCTR = 0x0000 || TBCTR = TBPRD)
- Time-base counter equal to the compare A register (CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is decrementing.

The number of events that have occurred can be read from the interrupt event counter (ETPS[INTCNT]) register bits. That is, when the specified event occurs the ETPS[INTCNT] bits are incremented until they reach the value specified by ETPS[INTPRD]. When ETPS[INTCNT] = ETPS[INTPRD] the counter stops counting and its output is set. The counter is only cleared when an interrupt is sent to the VIM.

When ETPS[INTCNT] reaches ETPS[INTPRD] the following behaviors will occur:

- If interrupts are enabled, ETSEL[INTEN] = 1 and the interrupt flag is clear, ETFLG[INT] = 0, then an interrupt pulse is generated and the interrupt flag is set, ETFLG[INT] = 1, and the event counter is cleared ETPS[INTCNT] = 0. The counter will begin counting events again.
- If interrupts are disabled, ETSEL[INTEN] = 0, or the interrupt flag is set, ETFLG[INT] = 1, the counter stops counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD].
- If interrupts are enabled, but the interrupt flag is already set, then the counter will hold its output high until the ENTFLG[INT] flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing to the INTPRD bits will automatically clear the counter INTCNT = 0 and the counter output will be reset (so no interrupts are generated). Writing a 1 to the ETFRC[INT] bit will increment the event counter INTCNT. The counter will behave as described above when INTCNT = INTPRD. When INTPRD = 0, the counter is disabled and hence no events will be detected and the ETFRC[INT] bit is also ignored.

The above definition means that you can generate an interrupt on every event, on every second event, or on every third event. An interrupt cannot be generated on every fourth or more events.

Figure 20-40. Event-Trigger Interrupt Generator

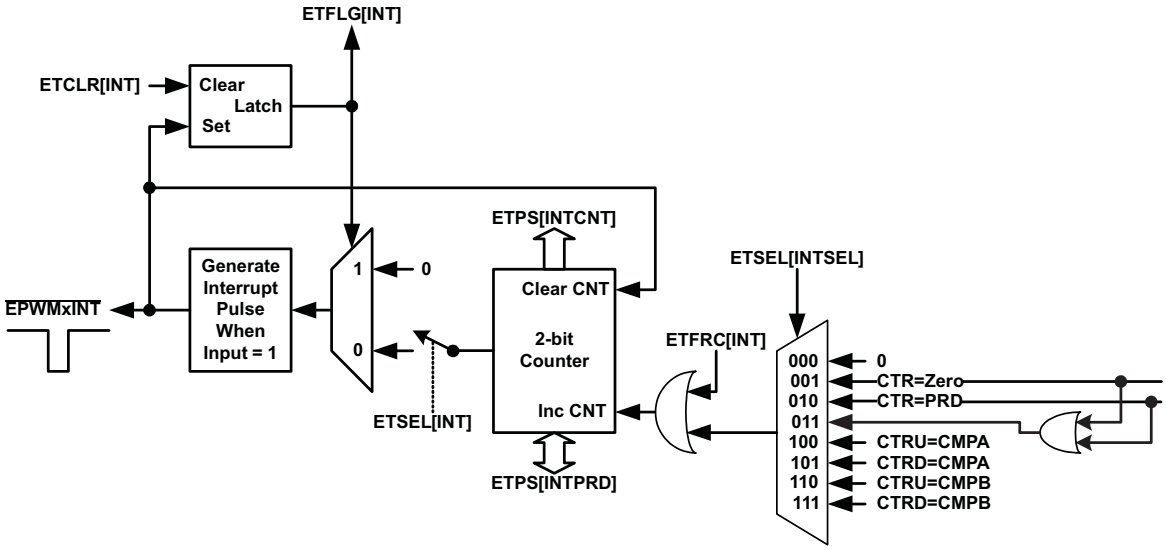
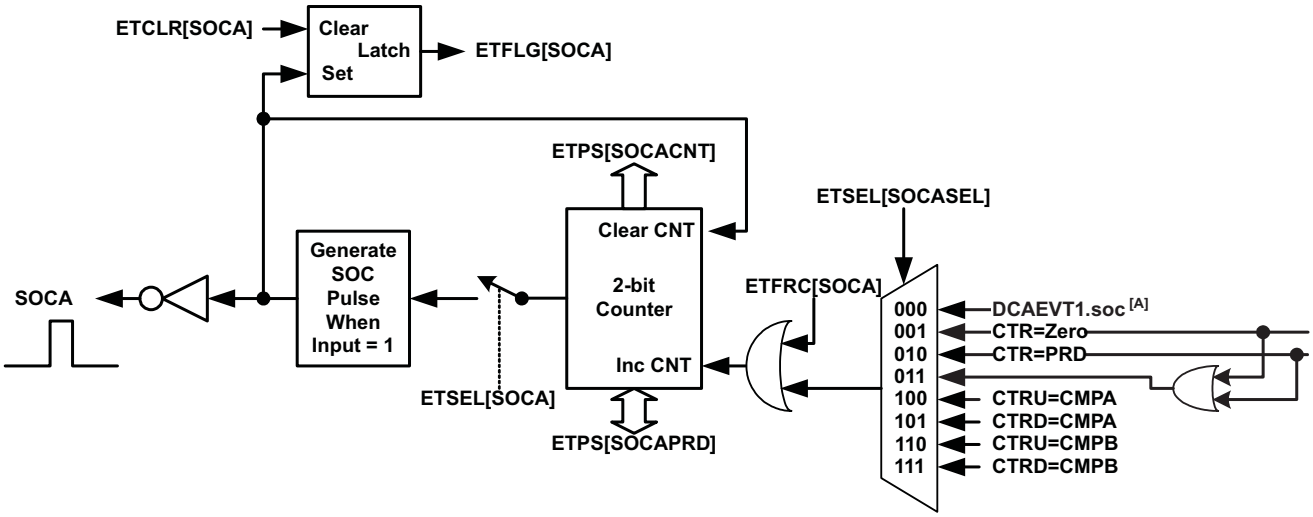


Figure 20-41 shows the operation of the event-trigger's start-of-conversion-A (SOCA) pulse generator. The ETPS[SOCACNT] counter and ETPS[SOCAPRD] period values behave similarly to the interrupt generator except that the pulses are continuously generated. That is, the pulse flag ETFLG[SOCA] is latched when a pulse is generated, but it does not stop further pulse generation. The enable/disable bit ETSEL[SOCAEN] stops pulse generation, but input events can still be counted until the period value is reached as with the interrupt generation logic. The event that will trigger an SOCA and SOCB pulse can be configured separately in the ETSEL[SOCASEL] and ETSEL[SOCBSEL] bits. The possible events are the same events that can be specified for the interrupt generation logic with the addition of the DCAEVT1.soc and DCBEVT1.soc event signals from the digital compare (DC) submodule.

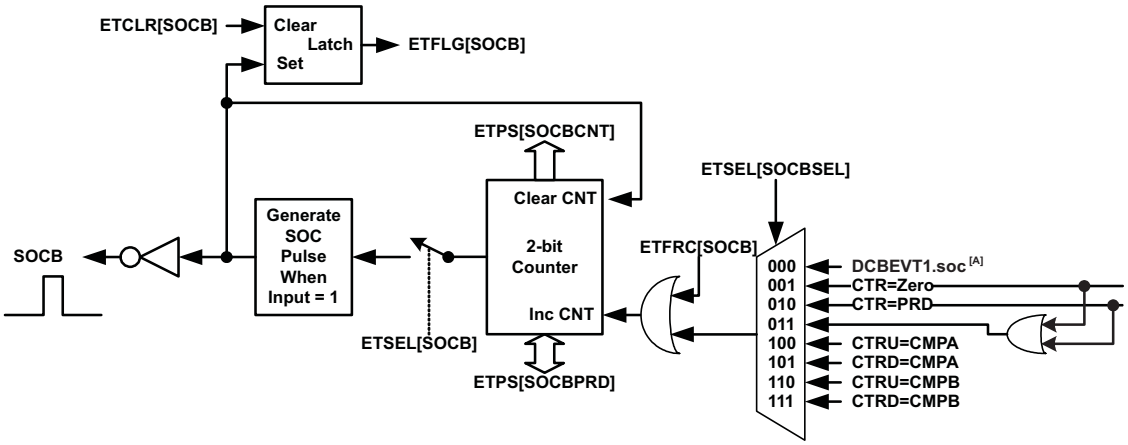
Figure 20-41. Event-Trigger SOCA Pulse Generator



A The DCAEVT1.soc signals are signals generated by the Digital compare (DC) submodule, described in Section 20.2.9

Figure 20-42 shows the operation of the event-trigger's start-of-conversion-B (SOCB) pulse generator. The event-trigger's SOCB pulse generator operates the same way as the SOCA.

Figure 20-42. Event-Trigger SOCB Pulse Generator



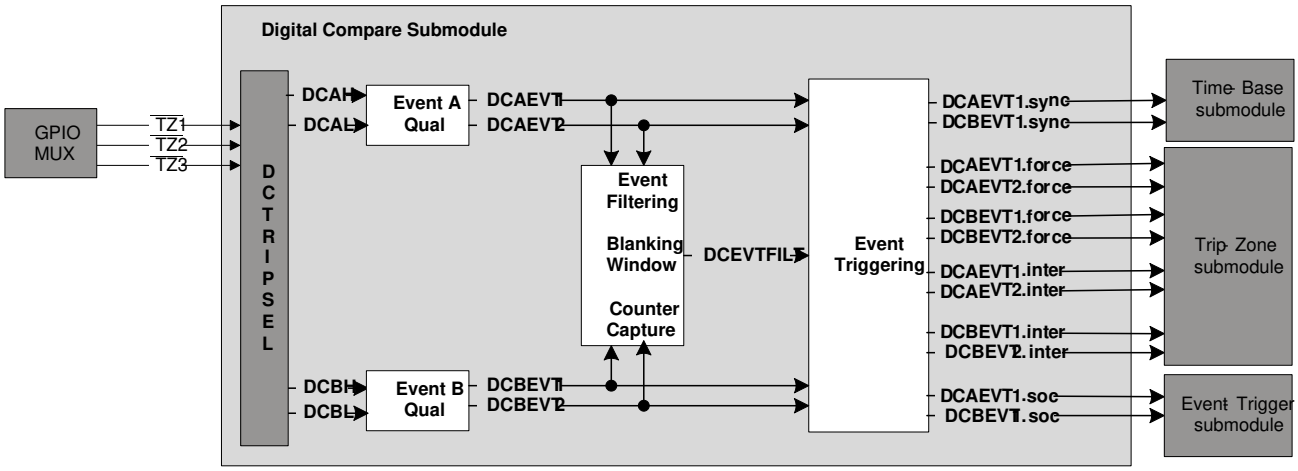
A The DCBEVT1.soc signals are signals generated by the Digital compare (DC) submodule, described in Section 20.2.9

20.2.9 Digital Compare (DC) Submodule

Figure 20-43 illustrates where the digital compare (DC) submodule signals interface to other submodules in the ePWM system.

The digital compare (DC) submodule compares signals external to the ePWM module to directly generate PWM events/actions which then feed to the event-trigger, trip-zone, and time-base submodules. Additionally, blanking window functionality is supported to filter noise or unwanted pulses from the DC event signals.

Figure 20-43. Digital-Compare Submodule High-Level Block Diagram



20.2.9.1 Purpose of the Digital Compare Submodule

The key functions of the digital compare submodule are:

- $\overline{TZ1}$, $\overline{TZ2}$, and $\overline{TZ3}$ inputs generate Digital Compare A High/Low (DCAH, DCAL) and Digital Compare B High/Low (DCBH, DCBL) signals.
- DCAH/L and DCBH/L signals trigger events which can then either be filtered or fed directly to the trip-zone, event-trigger, and time-base submodules to:
  - generate a trip zone interrupt
  - generate an ADC start of conversion
  - force an event
  - generate a synchronization event for synchronizing the ePWM module TBCTR.
- Event filtering (blanking window logic) can optionally blank the input signal to remove noise.

20.2.9.2 Controlling and Monitoring the Digital Compare Submodule

The digital compare submodule operation is controlled and monitored through the following registers:

Table 20-21. Digital Compare Submodule Registers

Register Name	Address Offset	Shadowed	Description
TZDCSEL[1] [2]		No	Trip Zone Digital Compare Select Register
DCACTL[1]		No	Digital Compare A Control Register
DCTRIPSEL[1]		No	Digital Compare Trip Select Register
DCFCTL[1]		No	Digital Compare Filter Control Register
DCBCTL[1]		No	Digital Compare B Control Register
DCFOFFSET		Writes	Digital Compare Filter Offset Register
DCCAPCTL[1]		No	Digital Compare Capture Control Register
DCFWINDOW		No	Digital Compare Filter Window Register
DCFOFFSETCNT		No	Digital Compare Filter Offset Counter Register
DCCAP		Yes	Digital Compare Counter Capture Register
DCFWINDOWCNT		No	Digital Compare Filter Window Counter Register

(1) These registers are writable only in privileged mode.

(2) The TZDCSEL register is part of the trip-zone submodule but is mentioned again here because of its functional significance to the digital compare submodule.

20.2.9.3 Operation Highlights of the Digital Compare Submodule

The following sections describe the operational highlights and configuration options for the digital compare submodule.

20.2.9.3.1 Digital Compare Events

As illustrated in Figure 20-43 earlier in this section, trip zone inputs ($\overline{TZ1}$, $\overline{TZ2}$, and $\overline{TZ3}$) can be selected via the DCTRIPSEL bits to generate the Digital Compare A High and Low (DCAH/L) and Digital Compare B High and Low (DCBH/L) signals. Then, the configuration of the TZDCSEL register qualifies the actions on the selected DCAH/L and DCBH/L signals, which generate the DCAEVT1/2 and DCBEVT1/2 events (Event Qualification A and B).

NOTE: The $\overline{TZn}$ signals, when used as a DCEVT tripping functions, are treated as a normal input signal and can be defined to be active high or active low inputs. EPWM outputs are asynchronously tripped when either the $\overline{TZn}$, DCAEVTx.force, or DCBEVTx.force signals are active. For the condition to remain latched, a minimum of 3*TBCLK sync pulse width is required. If pulse width is < 3*TBCLK sync pulse width, the trip condition may or may not get latched by CBC or OST latches.

The DCAEVT1/2 and DCBEVT1/2 events can then be filtered to provide a filtered version of the event signals (DCEVTFILT) or the filtering can be bypassed. Filtering is discussed further in section 2.9.3.2. Either the DCAEVT1/2 and DCBEVT1/2 event signals or the filtered DCEVTFILT event signals can generate a force to the trip zone module, a TZ interrupt, an ADC SOC, or a PWM sync signal.

- **force signal:**

  DCAEVT1/2.force signals force trip zone conditions which either directly influence the output on the EPWMxA pin (via TZCTL[DCAEVT1 or DCAEVT2] configurations) or, if the DCAEVT1/2 signals are selected as one-shot or cycle-by-cycle trip sources (via the TZSEL register), the DCAEVT1/2.force signals can effect the trip action via the TZCTL[TZA] configuration. The DCBEVT1/2.force signals behaves similarly, but affect the EPWMxB output pin instead of the EPWMxA output pin.

  The priority of conflicting actions on the TZCTL register is as follows (highest priority overrides lower priority):

  Output EPWMxA: TZA (highest) -> DCAEVT1 -> DCAEVT2 (lowest)

  Output EPWMxB: TZB (highest) -> DCBEVT1 -> DCBEVT2 (lowest)

- **interrupt signal:**

  DCAEVT1/2.interrupt signals generate trip zone interrupts to the VIM. To enable the interrupt, the user must set the DCAEVT1, DCAEVT2, DCBEVT1, or DCBEVT2 bits in the TZEINT register. Once one of these events occurs, an EPWMxTZINT interrupt is triggered, and the corresponding bit in the TZCLR register must be set in order to clear the interrupt.

- **soc signal:**

  The DCAEVT1.soc signal interfaces with the event-trigger submodule and can be selected as an event which generates an ADC start-of-conversion-A (SOCA) pulse via the ETSEL[SOCASEL] bit. Likewise, the DCBEVT1.soc signal can be selected as an event which generates an ADC start-of-conversion-B (SOCB) pulse via the ETSEL[SOCBSEL] bit.

- **sync signal:**

  The DCAEVT1.sync and DCBEVT1.sync events are ORed with the EPWMxSYNCI input signal and the TBCTL[SWFSYNC] signal to generate a synchronization pulse to the time-base counter.

Figure 20-44 and Figure 20-45 show how the DCAEVT1, DCAEVT2, or DCEVTFILT signals are processed to generate the digital compare A event force, interrupt, soc and sync signals.

Figure 20-44. DCAEVT1 Event Triggering

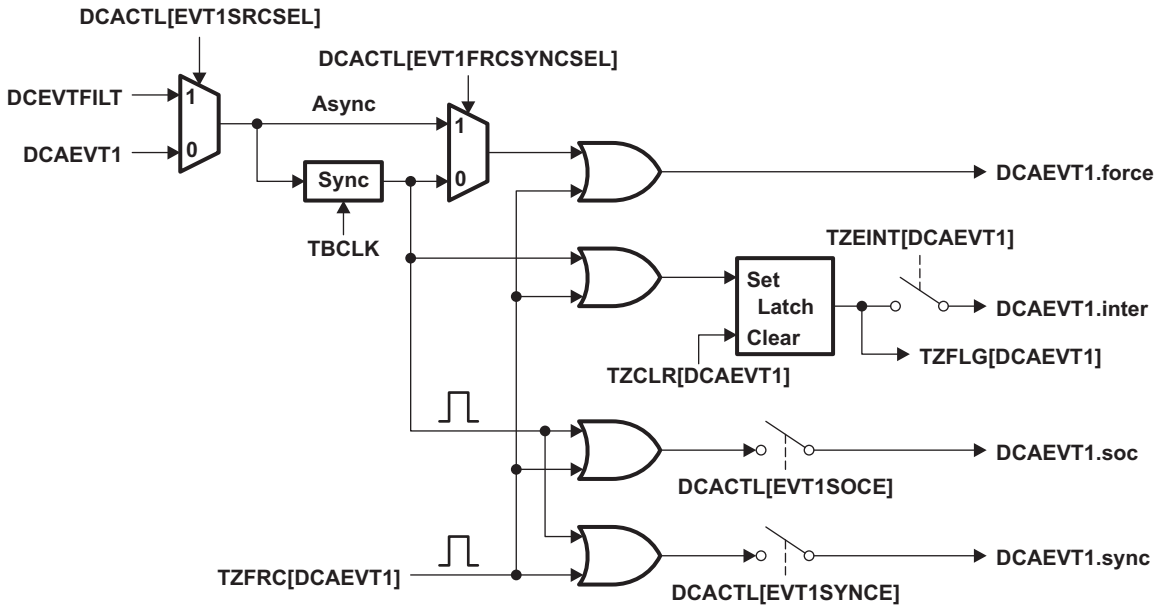


Figure 20-45. DCAEVT2 Event Triggering

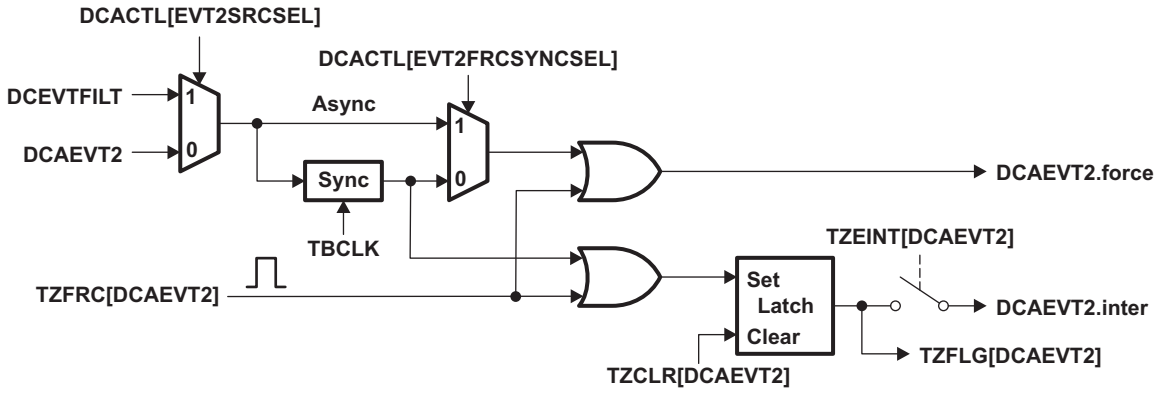


Figure 20-46 and Figure 20-47 show how the DCBEVT1, DCBEVT2, or DCEVTFILT signals are processed to generate the digital compare B event force, interrupt, soc and sync signals.

Figure 20-46. DCBEVT1 Event Triggering

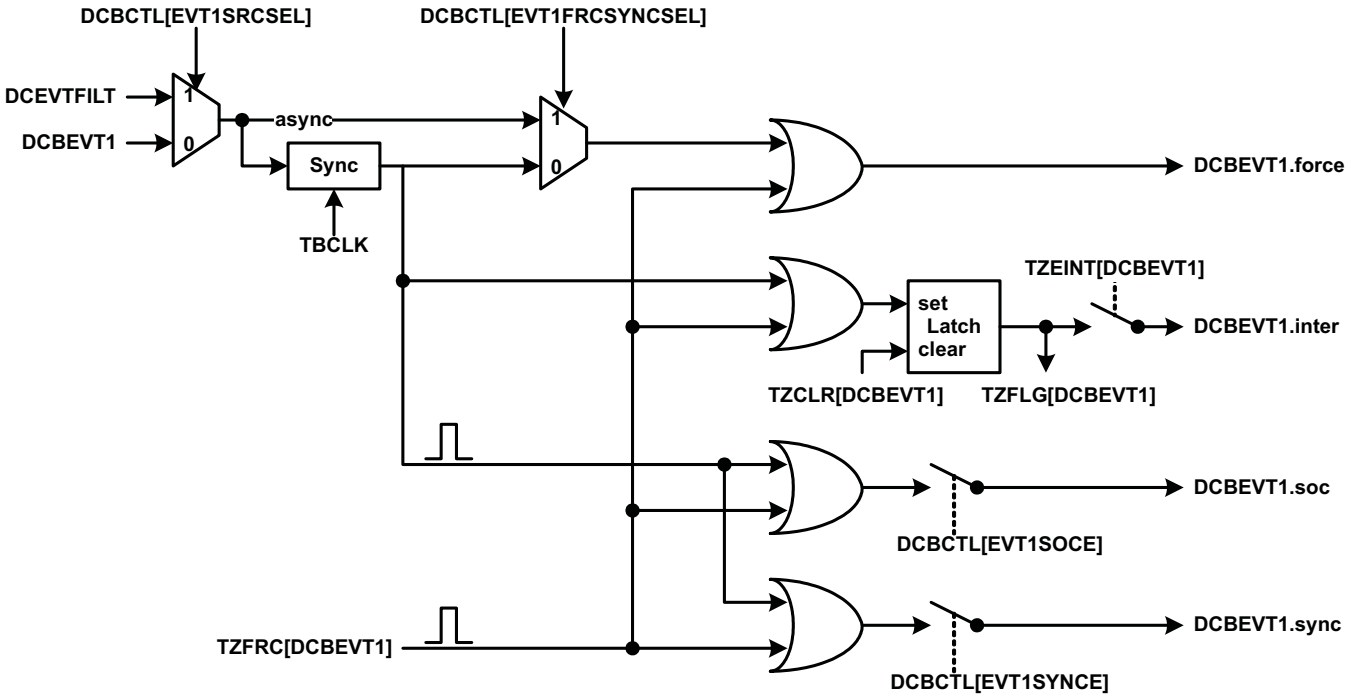
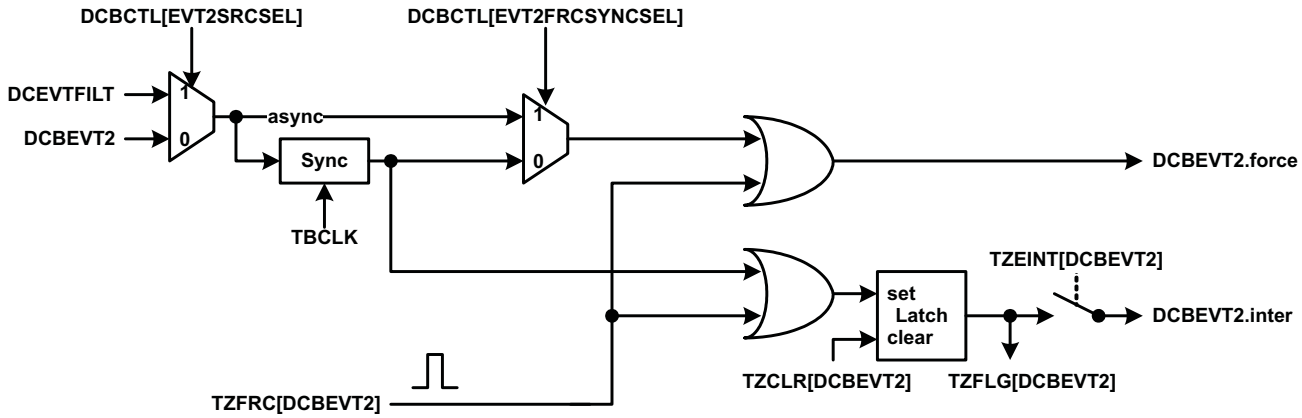


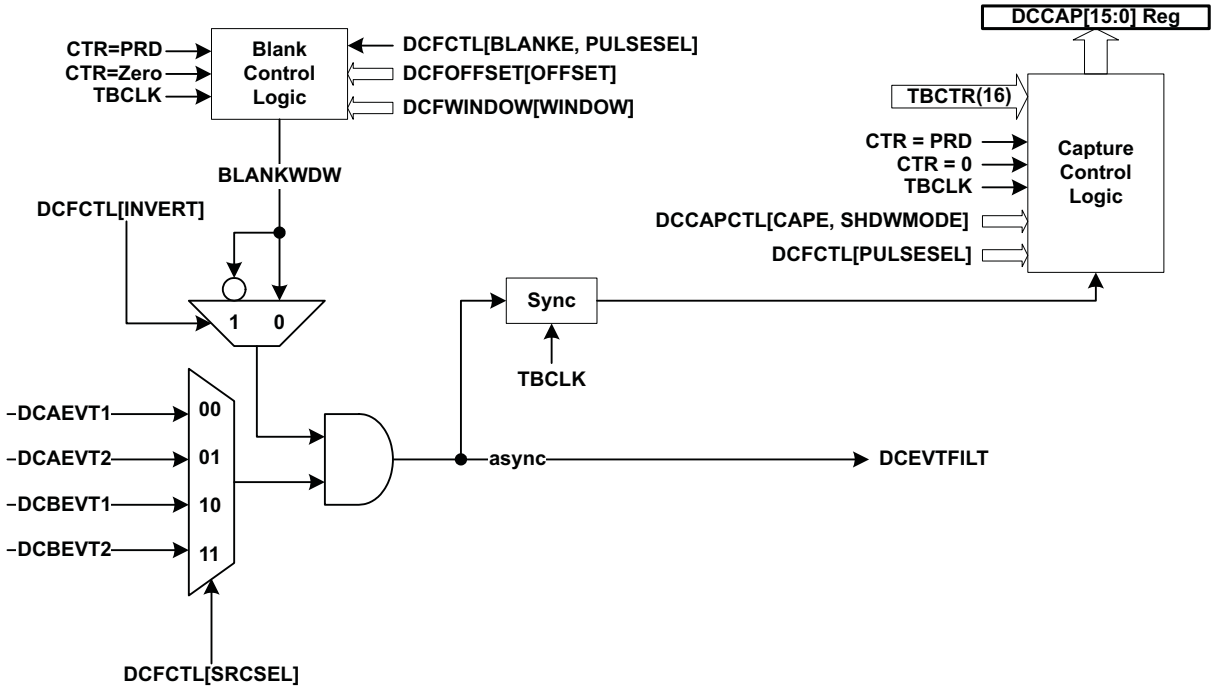
Figure 20-47. DCBEVT2 Event Triggering



20.2.9.3.2 Event Filtering

The DCAEVT1/2 and DCBEVT1/2 events can be filtered via event filtering logic to remove noise by optionally blanking events for a certain period of time. This is useful for cases where the analog comparator outputs may be selected to trigger DCAEVT1/2 and DCBEVT1/2 events, and the blanking logic is used to filter out potential noise on the signal prior to tripping the PWM outputs or generating an interrupt or ADC start-of-conversion. The event filtering can also capture the TBCTR value of the trip event. Figure 20-48 shows the details of the event filtering logic.

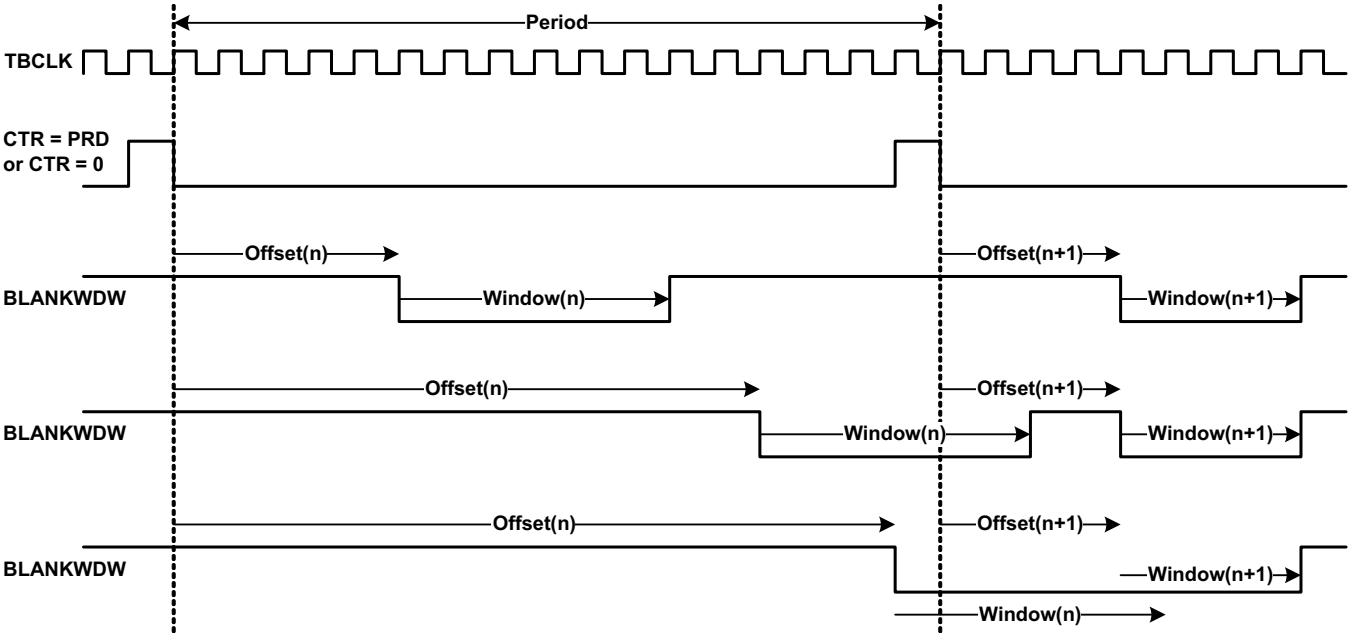
Figure 20-48. Event Filtering



If the blanking logic is enabled, one of the digital compare events – DCAEVT1, DCAEVT2, DCBEVT1, DCBEVT2 – is selected for filtering. The blanking window, which filters out all event occurrences on the signal while it is active, will be aligned to either a CTR = PRD pulse or a CTR = 0 pulse (configured by the DCFCTL[PULSESEL] bits). An offset value in TBCLK counts is programmed into the DCFOFFSET register, which determines at what point after the CTR = PRD or CTR = 0 pulse the blanking window starts. The duration of the blanking window, in number of TBCLK counts after the offset counter expires, is written to the DCFWINDOW register by the application. During the blanking window, all events are ignored. Before and after the blanking window ends, events can generate soc, sync, interrupt, and force signals as before.

Figure 20-49 shows several timing conditions for the offset and blanking window within an ePWM period. Notice that if the blanking window crosses the CTR = 0 or CTR = PRD boundary, the next window still starts at the same offset value after the CTR = 0 or CTR = PRD pulse.

Figure 20-49. Blanking Window Timing Diagram



20.2.10 Proper Interrupt Initialization Procedure

When the ePWM peripheral clock is enabled it may be possible that interrupt flags may be set due to spurious events due to the ePWM registers not being properly initialized. The proper procedure for initializing the ePWM peripheral is as follows:

1. Disable global interrupts (CPU INTM flag)
2. Disable ePWM interrupts
3. Set TBCLKSYNC = 0
4. Initialize peripheral registers
5. Set TBCLKSYNC = 1
6. Clear any spurious ePWM flags (including interrupt flags)
7. Enable ePWM interrupts
8. Enable global interrupts

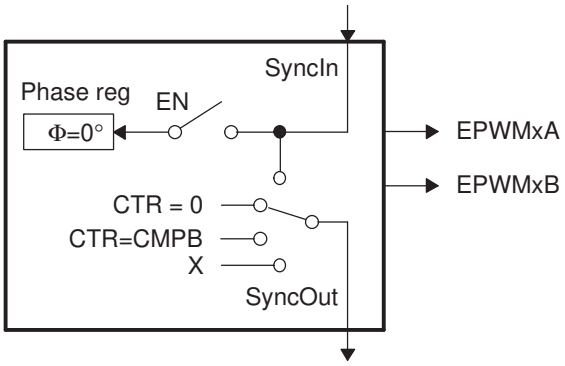
20.3 Application Examples

An ePWM module has all the local resources necessary to operate completely as a standalone module or to operate in synchronization with other identical ePWM modules.

20.3.1 Overview of Multiple Modules

Previously in this user's guide, all discussions have described the operation of a single module. To facilitate the understanding of multiple modules working together in a system, the ePWM module described in reference is represented by the more simplified block diagram shown in Figure 20-50. This simplified ePWM block shows only the key resources needed to explain how a multiswitch power topology is controlled with multiple ePWM modules working together.

Figure 20-50. Simplified ePWM Module



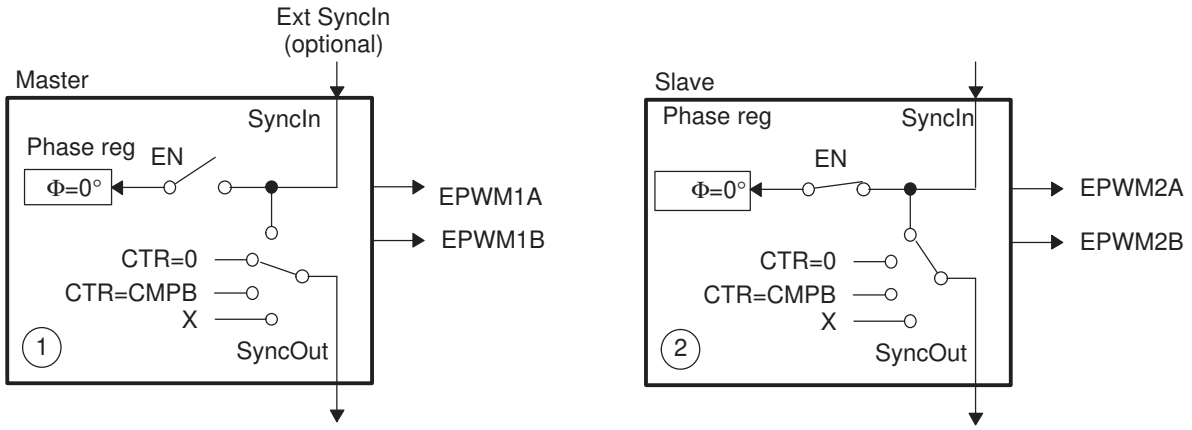
20.3.2 Key Configuration Capabilities

The key configuration choices available to each module are as follows:

- Options for SyncIn
  - Load own counter with phase register on an incoming sync strobe—enable (EN) switch closed
  - Do nothing or ignore incoming sync strobe—enable switch open
  - Sync flow-through - SyncOut connected to SyncIn
  - Master mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Master mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)
- Options for SyncOut
  - Sync flow-through - SyncOut connected to SyncIn
  - Master mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Master mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)

For each choice of SyncOut, a module may also choose to load its own counter with a new phase value on a SyncIn strobe input or choose to ignore it, i.e., via the enable switch. Although various combinations are possible, the two most common—master module and slave module modes—are shown in Figure 20-51.

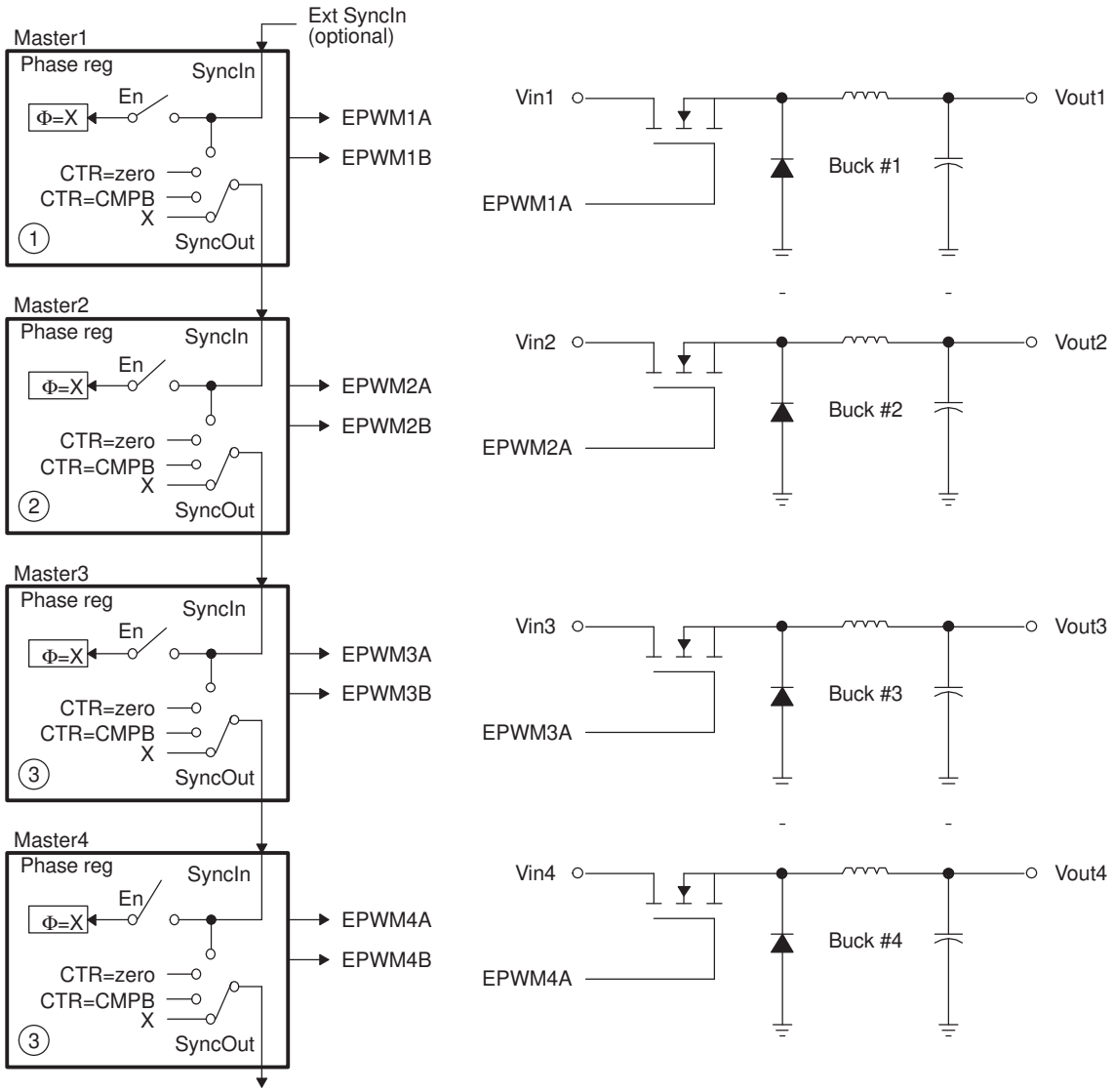
Figure 20-51. EPWM1 Configured as a Typical Master, EPWM2 Configured as a Slave



20.3.3 Controlling Multiple Buck Converters With Independent Frequencies

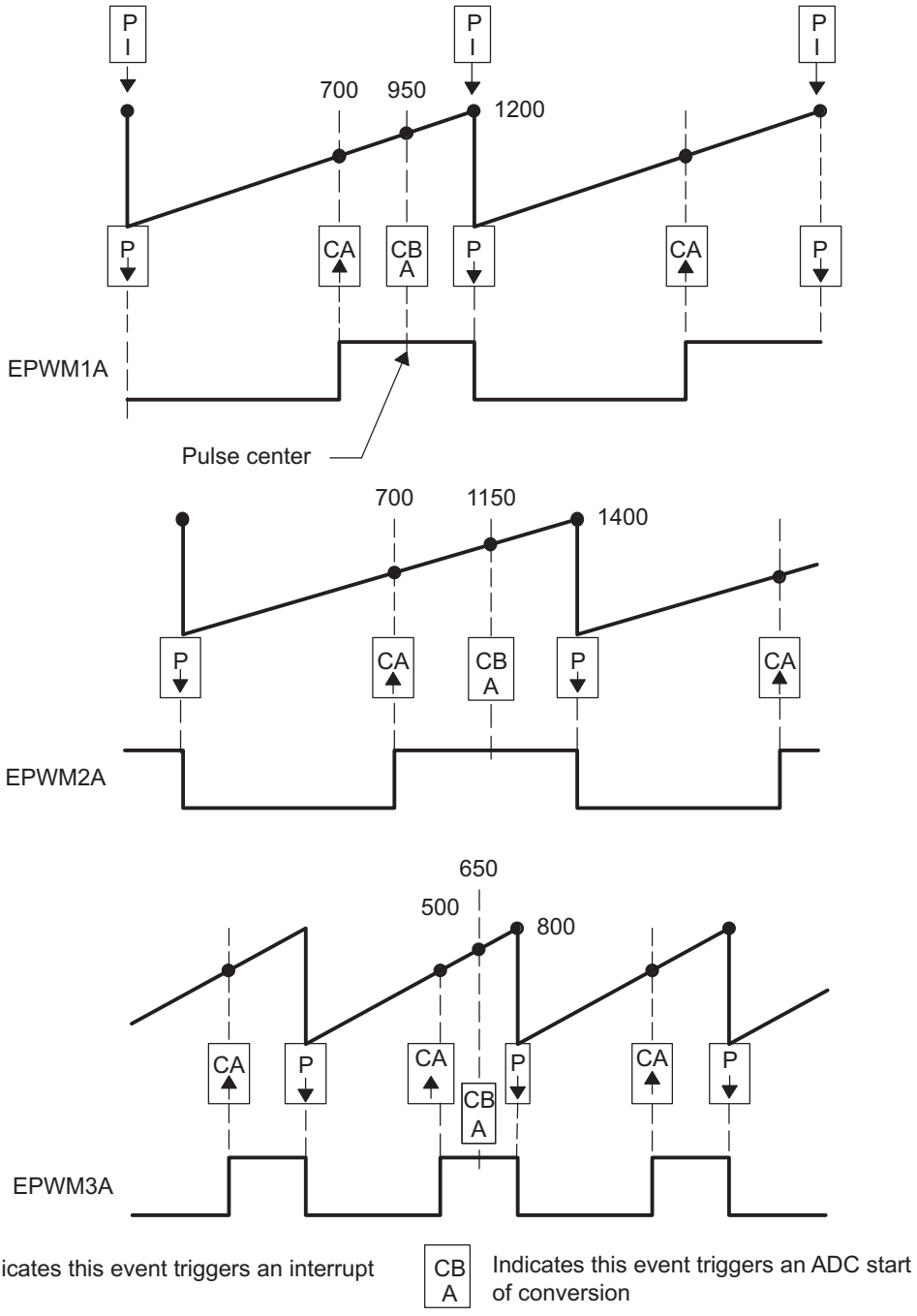
One of the simplest power converter topologies is the buck. A single ePWM module configured as a master can control two buck stages with the same PWM frequency. If independent frequency control is required for each buck converter, then one ePWM module must be allocated for each converter stage. Figure 20-52 shows four buck stages, each running at independent frequencies. In this case, all four ePWM modules are configured as Masters and no synchronization is used. Figure 20-53 shows the waveforms generated by the setup shown in Figure 20-52; note that only three waveforms are shown, although there are four stages.

Figure 20-52. Control of Four Buck Stages. Here $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$



NOTE: Θ = X indicates value in phase register is a "don't care"

Figure 20-53. Buck Waveforms for Figure 20-52 (Note: Only three bucks shown here)



Example 20-8. Configuration for Example in Figure 20-53

```
//=====================================================================
// (Note: code for only 3 modules shown)
// Initialization Time
//=======================
// EPWM Module 1 config
   EPwm1Regs.TBPRD = 1200;                          // Period = 1201 TBCLK counts
   EPwm1Regs.TBPHS.half.TBPHS = 0;                  // Set Phase register to zero
   EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;       // Asymmetrical mode
   EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;          // Phase loading disabled
   EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
   EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
   EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
   EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
   EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;    // load on CTR=Zero
   EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;    // load on CTR=Zero
   EPwm1Regs.AQCTLA.bit.PRD = AQ_CLEAR;
   EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
// EPWM Module 2 config
   EPwm2Regs.TBPRD = 1400;                          // Period = 1401 TBCLK counts
   EPwm2Regs.TBPHS.half.TBPHS = 0;                  // Set Phase register to zero
   EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;       // Asymmetrical mode
   EPwm2Regs.TBCTL.bit.PHSEN = TB_DISABLE;          // Phase loading disabled
   EPwm2Regs.TBCTL.bit.PRDLD = TB_SHADOW;
   EPwm2Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
   EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
   EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
   EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;    // load on CTR=Zero
   EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;    // load on CTR=Zero
   EPwm2Regs.AQCTLA.bit.PRD = AQ_CLEAR;
   EPwm2Regs.AQCTLA.bit.CAU = AQ_SET;
// EPWM Module 3 config
   EPwm3Regs.TBPRD = 800;                           // Period = 801 TBCLK counts
   EPwm3Regs.TBPHS.half.TBPHS = 0;                  // Set Phase register to zero
   EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
   EPwm3Regs.TBCTL.bit.PHSEN = TB_DISABLE;          // Phase loading disabled
   EPwm3Regs.TBCTL.bit.PRDLD = TB_SHADOW;
   EPwm3Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE;
   EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
   EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
   EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;    // load on CTR=Zero
   EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;    // load on CTR=Zero
   EPwm3Regs.AQCTLA.bit.PRD = AQ_CLEAR;
   EPwm3Regs.AQCTLA.bit.CAU = AQ_SET;
//
// Run Time (Note: Example execution of one run-time instant)
//=====================================================
   EPwm1Regs.CMPA.half.CMPA = 700;                  // adjust duty for output EPWM1A
   EPwm2Regs.CMPA.half.CMPA = 700;                  // adjust duty for output EPWM2A
   EPwm3Regs.CMPA.half.CMPA = 500;                  // adjust duty for output EPWM3A
```

20.3.4 Controlling Multiple Buck Converters With Same Frequencies

If synchronization is a requirement, ePWM module 2 can be configured as a slave and can operate at integer multiple (N) frequencies of module 1. The sync signal from master to slave ensures these modules remain locked. Figure 20-54 shows such a configuration; Figure 20-55 shows the waveforms generated by the configuration.

Figure 20-54. Control of Four Buck Stages. (Note: F_{PWM2} = N x F_{PWM1})

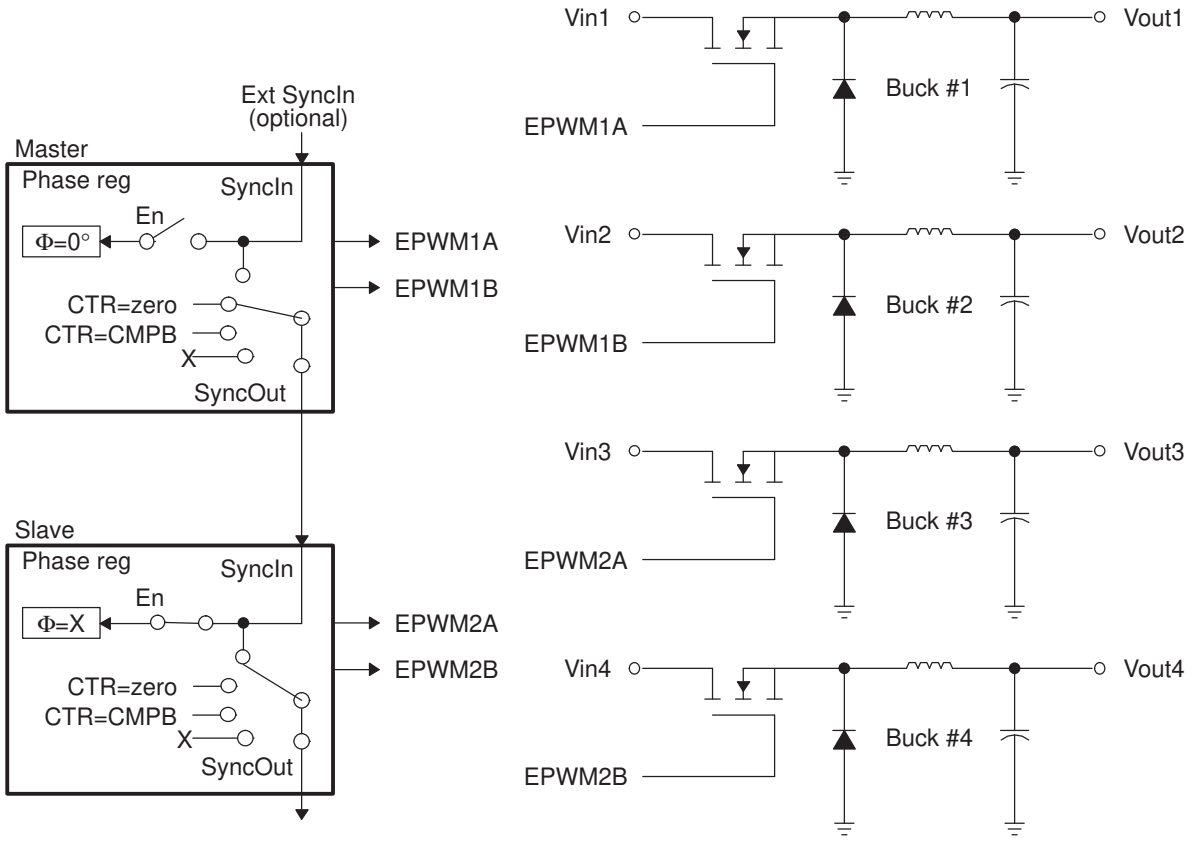
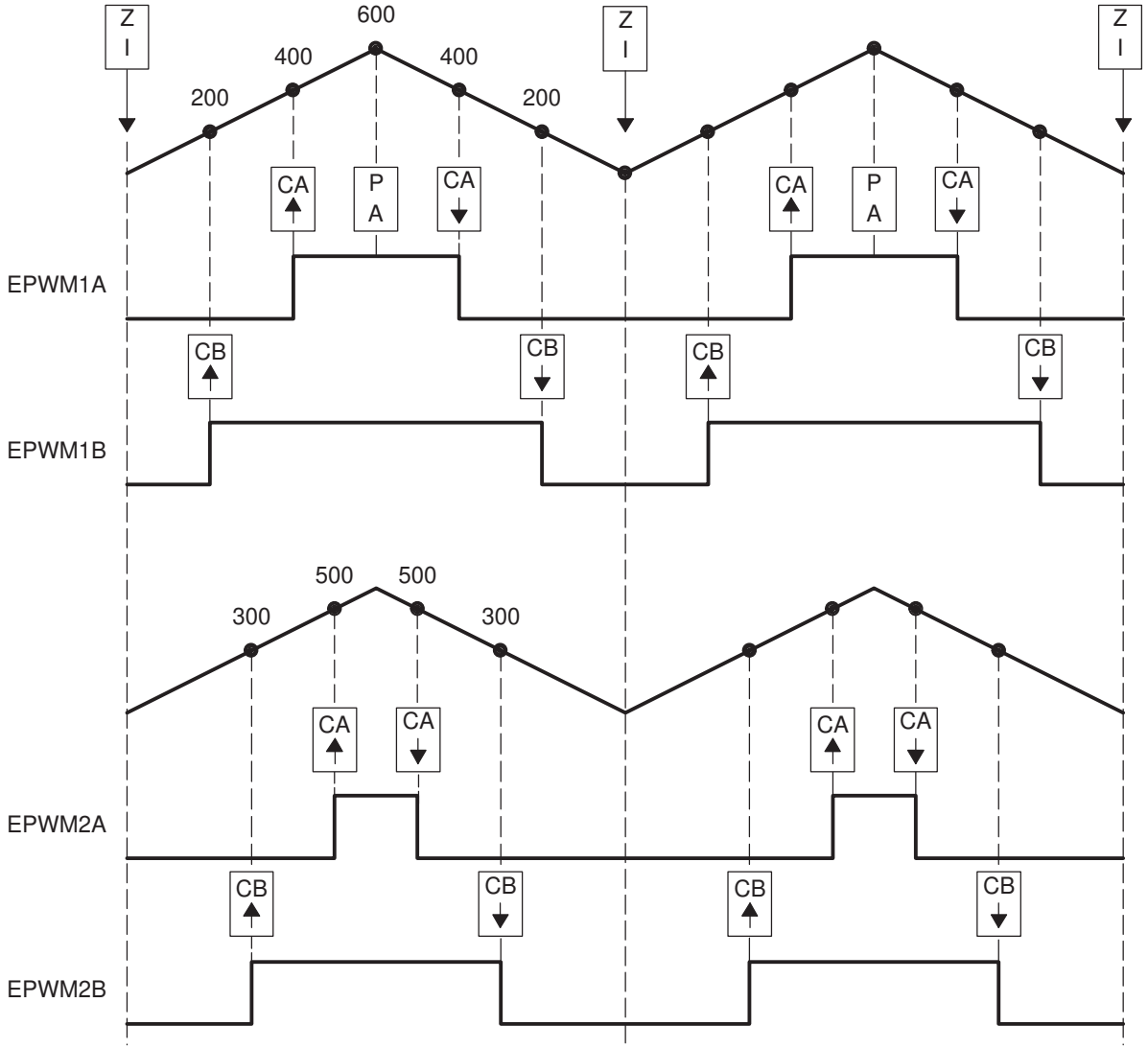


Figure 20-55. Buck Waveforms for Figure 20-54 (Note: $F_{PWM2} = F_{PWM1)}$)



Example 20-9. Code Snippet for Configuration in Figure 20-54

```
//=======================
// EPWM Module 1 config
     EPwm1Regs.TBPRD = 600;                        // Period = 1200 TBCLK counts
     EPwm1Regs.TBPHS.half.TBPHS = 0;               // Set Phase register to zero
     EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
     EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;       // Master module
     EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
     EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_CTR_ZERO;   // Sync down-stream module
     EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
     EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
     EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;            // set actions for EPWM1A
     EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
     EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;            // set actions for EPWM1B
     EPwm1Regs.AQCTLB.bit.CBD = AQ_CLEAR;
// EPWM Module 2 config
     EPwm2Regs.TBPRD = 600;                        // Period = 1200 TBCLK counts
     EPwm2Regs.TBPHS.half.TBPHS = 0;               // Set Phase register to zero
     EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
     EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE;        // Slave module
     EPwm2Regs.TBCTL.bit.PRDLD = TB_SHADOW;
     EPwm2Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_IN;    // sync flow-through
     EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
     EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
     EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm2Regs.AQCTLA.bit.CAU = AQ_SET;            // set actions for EPWM2A
     EPwm2Regs.AQCTLA.bit.CAD = AQ_CLEAR;
     EPwm2Regs.AQCTLB.bit.CBU = AQ_SET;            // set actions for EPWM2B
     EPwm2Regs.AQCTLB.bit.CBD = AQ_CLEAR;
//
// Run Time (Note: Example execution of one run-time instance)
//=====================================================
    EPwm1Regs.CMPA.half.CMPA = 400;               // adjust duty for output EPWM1A
    EPwm1Regs.CMPB = 200;                         // adjust duty for output EPWM1B
    EPwm2Regs.CMPA.half.CMPA = 500;               // adjust duty for output EPWM2A
    EPwm2Regs.CMPB = 300;                         // adjust duty for output EPWM2B
```

20.3.5 Controlling Multiple Half H-Bridge (HHB) Converters

Topologies that require control of multiple switching elements can also be addressed with these same ePWM modules. It is possible to control a Half-H bridge stage with a single ePWM module. This control can be extended to multiple stages. Figure 20-56 shows control of two synchronized Half-H bridge stages where stage 2 can operate at integer multiple (N) frequencies of stage 1. Figure 20-57 shows the waveforms generated by the configuration shown in Figure 20-56.

Module 2 (slave) is configured for Sync flow-through; if required, this configuration allows for a third Half-H bridge to be controlled by PWM module 3 and also, most importantly, to remain in synchronization with master module 1.

Figure 20-56. Control of Two Half-H Bridge Stages (F_{PWM2} = N x F_{PWM1})

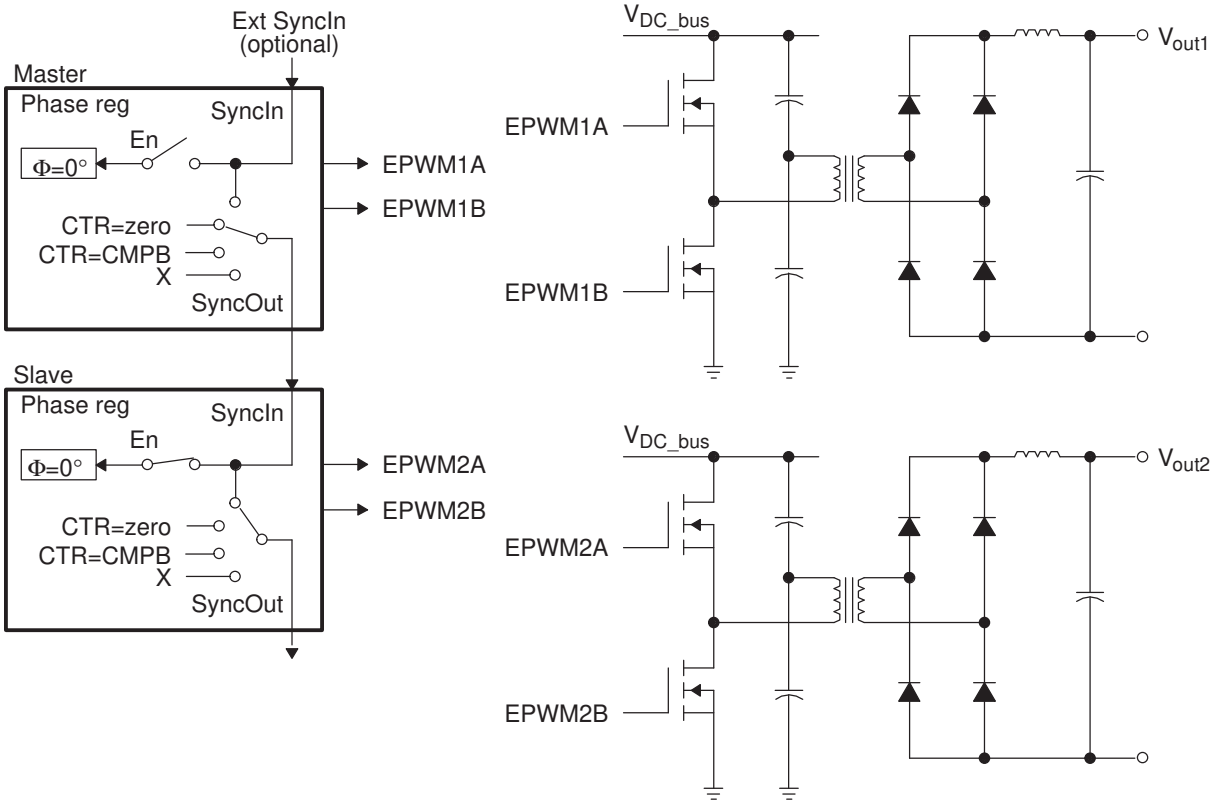
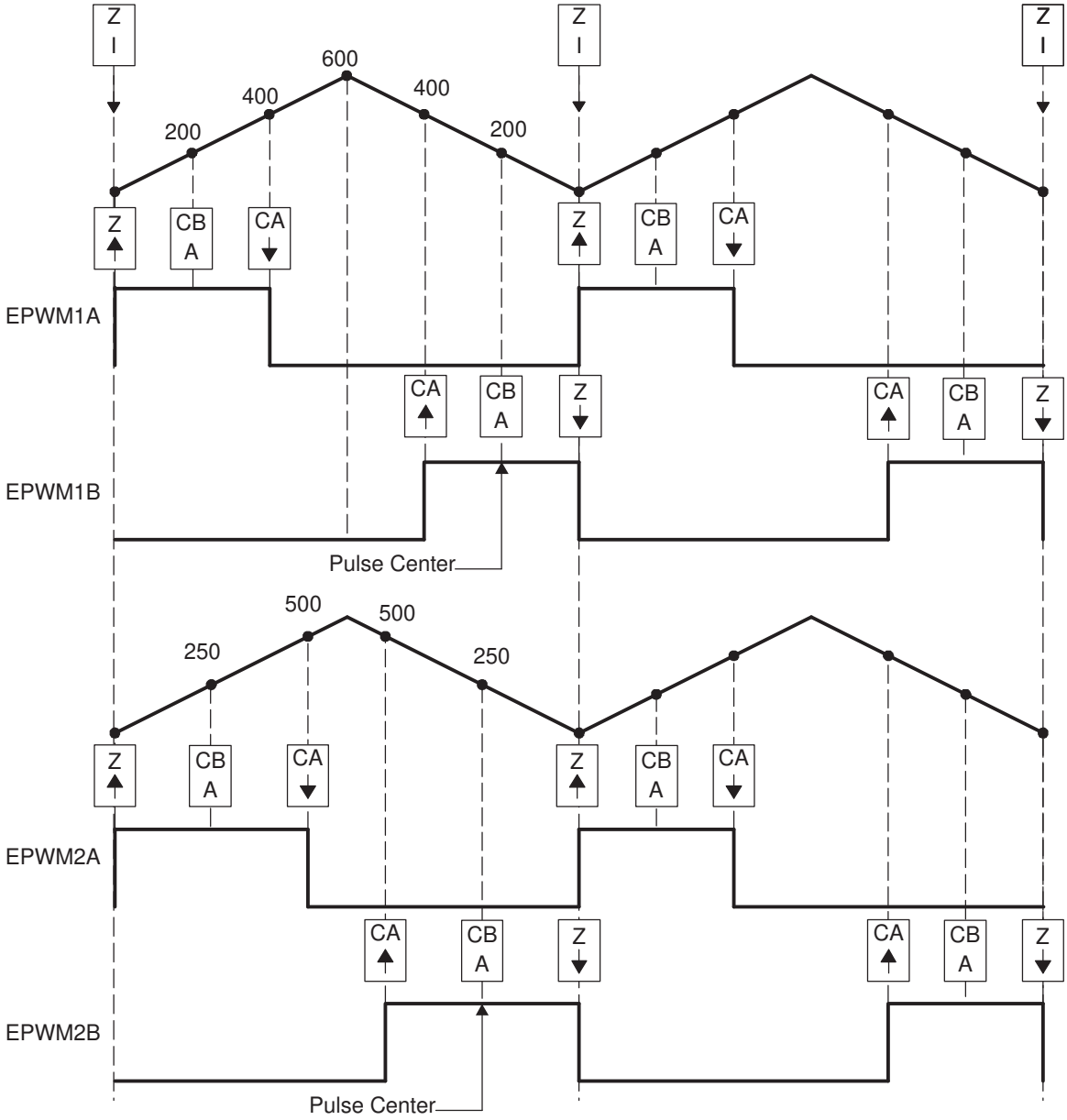


Figure 20-57. Half-H Bridge Waveforms for Figure 20-56 (Note: Here $F_{PWM2} = F_{PWM1}$)



Example 20-10. Code Snippet for Configuration in Figure 20-56

```
//=============================================================================
// Config
//=============================================================================
// Initialization Time
//=======================
// EPWM Module 1 config
     EPwm1Regs.TBPRD = 600;                        // Period = 1200 TBCLK counts
     EPwm1Regs.TBPHS.half.TBPHS = 0;               // Set Phase register to zero
     EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
     EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;       // Master module
     EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;
     EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_CTR_ZERO;   // Sync down-stream module
     EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
     EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
     EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;            // set actions for EPWM1A
     EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
     EPwm1Regs.AQCTLB.bit.ZRO = AQ_CLEAR;          // set actions for EPWM1B
     EPwm1Regs.AQCTLB.bit.CAD = AQ_SET;
// EPWM Module 2 config
     EPwm2Regs.TBPRD = 600;                        // Period = 1200 TBCLK counts
     EPwm2Regs.TBPHS.half.TBPHS = 0;               // Set Phase register to zero
     EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
     EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE;        // Slave module
     EPwm2Regs.TBCTL.bit.PRDLD = TB_SHADOW;
     EPwm2Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_IN;    // sync flow-through
     EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
     EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
     EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;  // load on CTR=Zero
     EPwm2Regs.AQCTLA.bit.ZRO = AQ_SET;            // set actions for EPWM1A
     EPwm2Regs.AQCTLA.bit.CAU = AQ_CLEAR;
     EPwm2Regs.AQCTLB.bit.ZRO = AQ_CLEAR;          // set actions for EPWM1B
     EPwm2Regs.AQCTLB.bit.CAD = AQ_SET;
//=============================================================
     EPwm1Regs.CMPA.half.CMPA = 400;               // adjust duty for output EPWM1A & EPWM1B

     EPwm1Regs.CMPB = 200;                         // adjust point-in-time for ADCSOC trigger
     EPwm2Regs.CMPA.half.CMPA = 500;               // adjust duty for output EPWM2A & EPWM2B
     EPwm2Regs.CMPB = 250;                         // adjust point-in-time for ADCSOC trigger
```

20.3.6 Controlling Dual 3-Phase Inverters for Motors (ACI and PMSM)

The idea of multiple modules controlling a single power stage can be extended to the 3-phase Inverter case. In such a case, six switching elements can be controlled using three PWM modules, one for each leg of the inverter. Each leg must switch at the same frequency and all legs must be synchronized. A master + two slaves configuration can easily address this requirement. Figure 20-58 shows how six PWM modules can control two independent 3-phase Inverters; each running a motor.

As in the cases shown in the previous sections, we have a choice of running each inverter at a different frequency (module 1 and module 4 are masters as in Figure 20-58), or both inverters can be synchronized by using one master (module 1) and five slaves. In this case, the frequency of modules 4, 5, and 6 (all equal) can be integer multiples of the frequency for modules 1, 2, 3 (also all equal).

Figure 20-58. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control

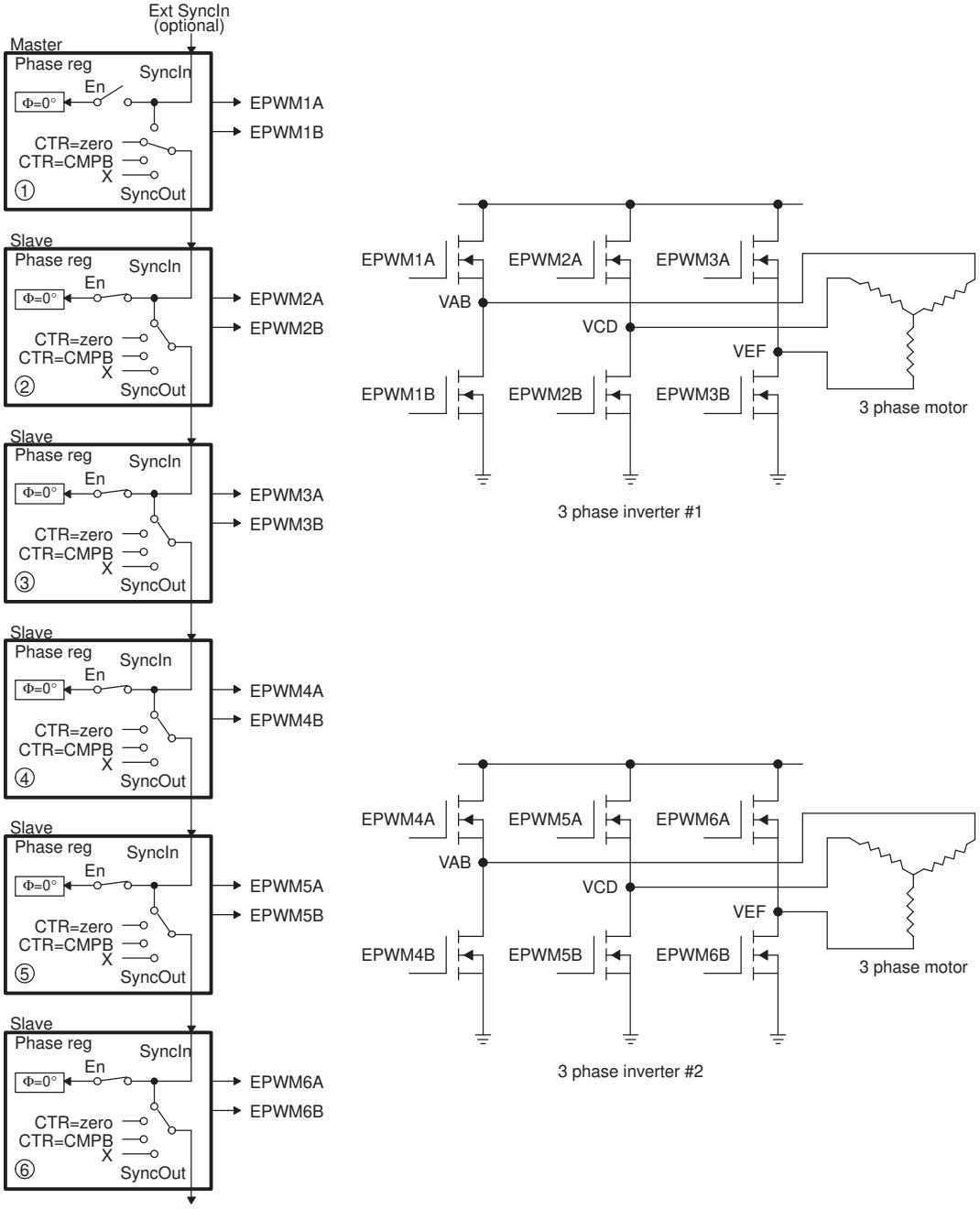
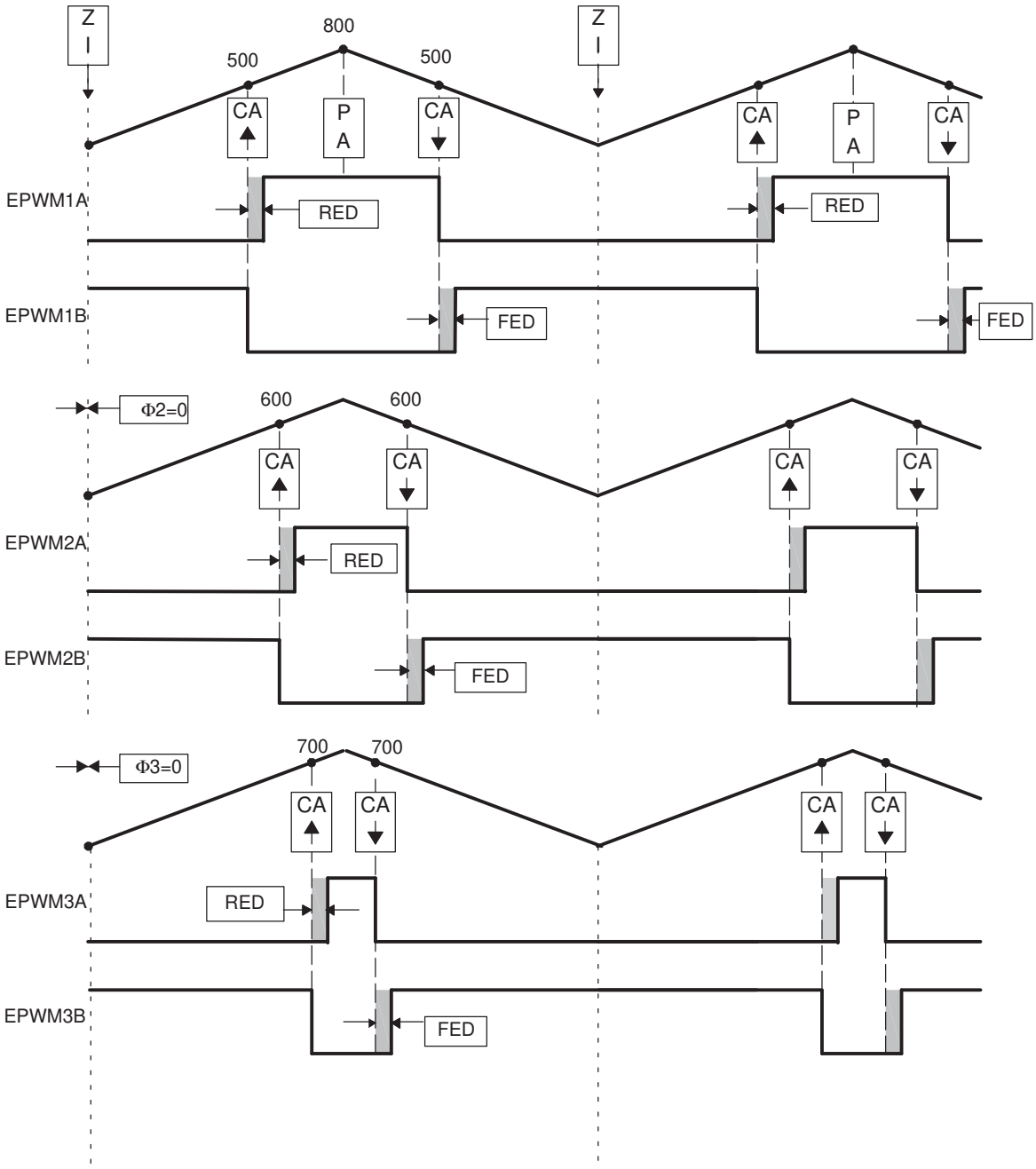


Figure 20-59. 3-Phase Inverter Waveforms for Figure 20-58 (Only One Inverter Shown)



20.3.7 Practical Applications Using Phase Control Between PWM Modules

So far, none of the examples have made use of the phase register (TBPHS). It has either been set to zero or its value has been a don't care. However, by programming appropriate values into TBPHS, multiple PWM modules can address another class of applications that rely on phase relationship between stages for correct operation. As described in the TB module section, a PWM module can be configured to allow a SyncIn pulse to cause the TBPHS register to be loaded into the TBCTR register. To illustrate this concept, Figure 20-60 shows a master and slave module with a phase relationship of 120°, i.e., the slave leads the master.

Figure 20-60. Configuring Two PWM Modules for Phase Control

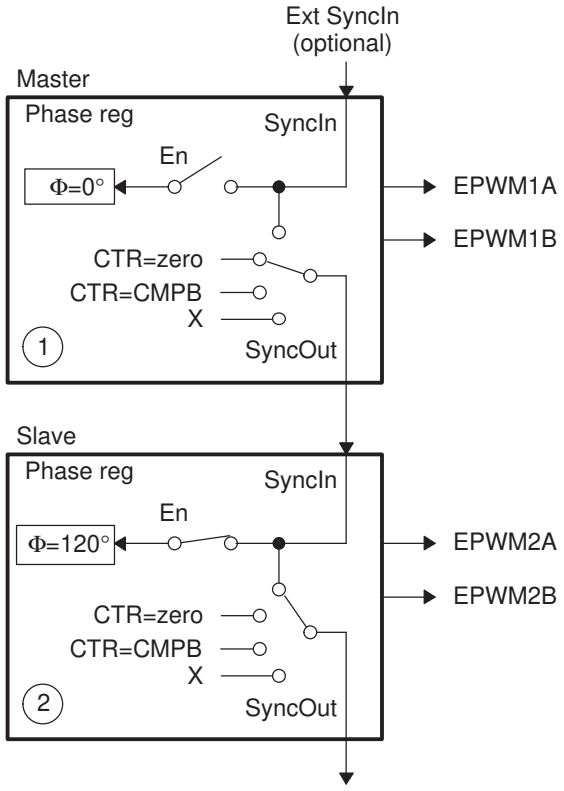
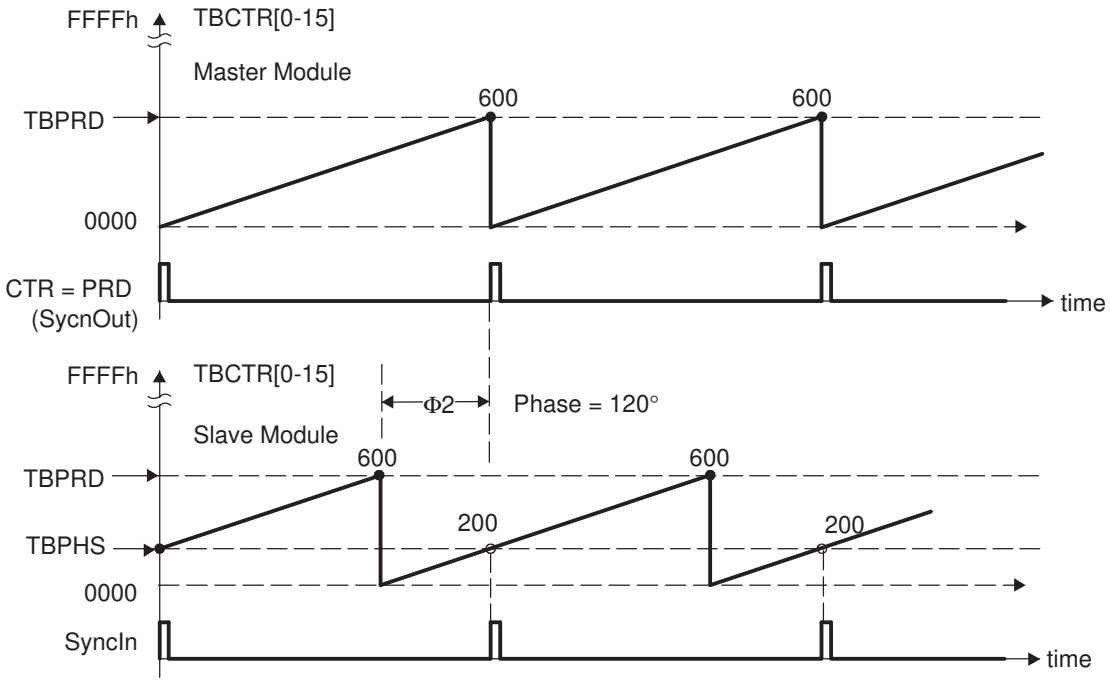


Figure 20-61 shows the associated timing waveforms for this configuration. Here, TBPRD = 600 for both master and slave. For the slave, TBPHS = 200 (200/600 × 360° = 120°). Whenever the master generates a SyncIn pulse (CTR = PRD), the value of TBPHS = 200 is loaded into the slave TBCTR register so the slave time-base is always leading the master's time-base by 120°.

Figure 20-61. Timing Waveforms Associated With Phase Control Between 2 Modules



20.4 MSS_ETPWM1 Registers

Table 20-22 lists the memory-mapped registers for the MSS_ETPWM1. Each register set is duplicated for each instance of the ePWM module. All register offset addresses not listed in Table 20-22 should be considered as reserved locations and the register contents should not be modified.

Table 20-22. MSS_ETPWM1 Registers

Offset	Acronym	Register Name	Section
0h	TBCTL_TBSTS	Time-Base Control Register/ Status Register	Section 20.4.1
4h	TBPHS	Time-Base Phase Register	Section 20.4.2
8h	TBCTR_TBPRD	Time-Base Counter Register/ Period Register	Section 20.4.3
Ch	CMPCTL	Counter-Compare Control Register	Section 20.4.4
10h	CMPA	Counter-Compare A Register	Section 20.4.5
14h	CMPB_AQCTLA	Counter-Compare B Register/ Action-Qualifier Control Register for Output A (EPWMxA)	Section 20.4.6
18h	AQCTLB_AQSFRC	Action-Qualifier Software Force Register/ Action-Qualifier Control Register for Output B (EPWMxB)	Section 20.4.7
1Ch	AQCSFRC_DBCTL	Dead-Band Generator Control Register/ Action-Qualifier Continuous S/W Force Register Set	Section 20.4.8
20h	DBRED_DBFED	Dead-Band Generator Rising Edge Delay Count Register/ Dead-Band Generator Falling Edge Delay Count Register	Section 20.4.9
24h	TZSEL_TZDCSEL	Trip Zone Digital Compare Select Register/ Trip-Zone Select Register	Section 20.4.10
28h	TZCTL_TZEINT	Trip-Zone Control Register/ Trip-Zone Enable Interrupt Register	Section 20.4.11
2Ch	TZFLG_TZCLR	Trip-Zone Flag Register/ Trip-Zone Clear Register	Section 20.4.12
30h	TZFRC_ETSEL	Trip-Zone Force Register / Event-Trigger Selection Register	Section 20.4.13
34h	ETPS_ETFLG	Event-Trigger Pre-Scale Register/ Event-Trigger Flag Register	Section 20.4.14
38h	ETCLR_ETFRC	Event-Trigger Clear Register/ Event-Trigger Force Register	Section 20.4.15
3Ch	PCCTL	PWM-Chopper Control Register	Section 20.4.16
40h	Reserved1	Reserved	Section 20.4.17
44h	Reserved2	Reserved	Section 20.4.18
48h	Reserved3	Reserved	Section 20.4.19
4Ch	Reserved4	Reserved	Section 20.4.20
50h	Reserved5	Reserved	Section 20.4.21
54h	Reserved6	Reserved	Section 20.4.22
58h	Reserved7	Reserved	Section 20.4.23
5Ch	Reserved8	Reserved	Section 20.4.24
60h	DCTRIPSEL_DCACTL	Digital Compare Trip Select Register/ Digital Compare A Control Register	Section 20.4.25
64h	DCBCTL_DCFCTL	Digital Compare B Control Register/ Digital Compare Filter Control Register	Section 20.4.26
68h	DCCAPCTL_DCFOFFSET	Digital Compare Capture Control Register/ Digital Compare Filter Offset Register	Section 20.4.27
6Ch	DCFOFFSETCNT_DCFWINDOW	Digital Compare Filter Offset Counter Register/ Digital Compare Filter Window Register	Section 20.4.28
70h	DCFWINDOWCNT_DCCAP	Digital Compare Filter Window Counter Register/ Digital Compare Counter Capture Register	Section 20.4.29

Complex bit access types are encoded to fit into small table cells. Table 20-23 shows the codes that are used for access types in this section.

Table 20-23. MSS_ETPWM1 Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

20.4.1 TBCTL_TBSTS Register (Offset = 0h) [reset = 0h]

TBCTL_TBSTS is shown in Figure 20-62 and described in Table 20-24.

Return to Summary Table.

Time-Base Control Register/ Status Register

Figure 20-62. TBCTL_TBSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					TBSTS_CTRMAX	TBSTS_SYNCI	TBSTS_CTRDIR
R-0h					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TBCTL_FREE,_SOFT		TBCTL_PHSDIR	TBCTL_CLKDIV			TBCTL_HSPCLKDIV	
R/W-0h		R/W-0h	R/W-0h			R/W-0h	
7	6	5	4	3	2	1	0
TBCTL_HSPCLKDIV	TBCTL_SWFSYNC	TBCTL_SYNCOSEL		TBCTL_PRDLD	TBCTL_PHSEN	TBCTL_CTRMODE	
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	

Table 20-24. TBCTL_TBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18	TBSTS_CTRMAX	R/W	0h	Time-Base Counter Max Latched Status Bit 0 Read: Indicates the time-base counter never reached its maximum value. Write: No effect. 1 Read: Indicates that the time-base counter reached the maximum value 0xFFFF. Write: Clears the latched event.
17	TBSTS_SYNCI	R/W	0h	Input Synchronization Latched Status Bit 0 Read: Indicates no external synchronization event has occurred. Write: No effect. 1 Read: Indicates that an external synchronization event has occurred (EPWMxSYNCI). Write: Clears the latched event.
16	TBSTS_CTRDIR	R/W	0h	Time-Base Counter Direction Status Bit. At reset, the counter is frozen; therefore, this bit has no meaning. To make this bit meaningful, you must first set the appropriate mode via TBCTL[CTRMODE]. 0 Time-Base Counter is currently counting down. 1 Time-Base Counter is currently counting up.
15-14	TBCTL_FREE,_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the ePWM time-base counter during emulation events: 0 Stop after the next time-base counter increment or decrement 1h Stop when counter completes a whole cycle: • Up-count mode: stop when the time-base counter = period (TBCTR = TBPRD) • Down-count mode: stop when the time-base counter = 0x0000 (TBCTR = 0x0000) • Up-down-count mode: stop when the time-base counter = 0x0000 (TBCTR = 0x0000) 2h-3h Free run
13	TBCTL_PHSDIR	R/W	0h	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. The PHSDIR bit indicates the direction the time-base counter (TBCTR) will count after a synchronization event occurs and a new phase value is loaded from the phase (TBPHS) register. This is irrespective of the direction of the counter before the synchronization event.. In the up-count and down-count modes this bit is ignored. 0 Count down after the synchronization event. 1 Count up after the synchronization event

Table 20-24. TBCTL_TBSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-10	TBCTL_CLKDIV	R/W	0h	Time-base Clock Prescale Bits. These bits determine part of the timebase clock prescale value. TBCLK = VCLK3 / (HSPCLKDIV * CLKDIV) 000: /1 (default on reset) 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
9-7	TBCTL_HSPCLKDIV	R/W	0h	High Speed Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value: TBCLK = VCLK3 / (HSPCLKDIV * CLKDIV) 000: /1 001: /2 (default on reset) 010: /4 011: /6 100: /8 101: /10 110: /12 111: /14
6	TBCTL_SWFSYNC	R/W	0h	Software Forced Synchronization Pulse 0 Writing a 0 has no effect and reads always return a 0. 1 Writing a 1 forces a one-time synchronization pulse to be generated. This event is ORed with the EPWMxSYNCI input of the ePWM module. SWFSYNC is valid (operates) only when EPWMxSYNCI is selected by SYNCOSEL = 00.
5-4	TBCTL_SYNCOSEL	R/W	0h	Synchronization Output Select. These bits select the source of the EPWMxSYNCO signal. 0 EPWMxSYNC: 1h CTR = zero: Time-base counter equal to zero (TBCTR = 0x0000) 2h CTR = CMPB : Time-base counter equal to counter-compare B (TBCTR = CMPB) 3h Disable EPWMxSYNCO signal
3	TBCTL_PRDLD	R/W	0h	Active Period Register Load From Shadow Register Select 0 The period register (TBPRD) is loaded from its shadow register when the time-base counter, TBCTR, is equal to zero. A write or read to the TBPRD register accesses the shadow register. 1 Load the TBPRD register immediately without using a shadow register. A write or read to the TBPRD register directly accesses the active register.
2	TBCTL_PHSEN	R/W	0h	Counter Register Load From Phase Register Enable 0 Do not load the time-base counter (TBCTR) from the time-base phase register (TBPHS) 1 Load the time-base counter with the phase register when an EPWMxSYNCI input signal occurs or when a software synchronization is forced by the SWFSYNC bit, or when a digital compare sync event occurs.
1-0	TBCTL_CTRMODE	R/W	0h	Counter Mode The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 0 Up-count mode 1h Down-count mode 2h Up-down-count mode 3h Stop-freeze counter operation (default on reset)

20.4.2 TBPHS Register (Offset = 4h) [reset = 0h]

TBPHS is shown in Figure 20-63 and described in Table 20-25.

Return to Summary Table.

Time-Base Phase Register

Figure 20-63. TBPHS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS																RESERVED															
R/W-0h																R-0h															

Table 20-25. TBPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TBPHS	R/W	0h	Time-Base Phase Register These bits set time-base counter phase of the selected ePWM relative to the time-base that is supplying the synchronization input signal. • If TBCTL[PHSEN] = 0, then the synchronization event is ignored and the time-base counter is not loaded with the phase. • If TBCTL[PHSEN] = 1, then the time-base counter (TBCTR) will be loaded with the phase (TBPHS) when a synchronization event occurs. The synchronization event can be initiated by the input synchronization signal (EPWMxSYNCI) or by a software forced synchronization.
15-0	RESERVED	R	0h	Reserved

20.4.3 TBCTR_TBPRD Register (Offset = 8h) [reset = 0h]

TBCTR_TBPRD is shown in Figure 20-64 and described in Table 20-26.

Return to Summary Table.

Time-Base Counter Register/ Period Register

Figure 20-64. TBCTR_TBPRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRD																TBCTR															
R/W-0h																R/W-0h															

Table 20-26. TBCTR_TBPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TBPRD	R/W	0h	Time-Base Period Register These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed. • If TBCTL[PRDLD] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the active register will be loaded from the shadow register when the timebase counter equals zero. • If TBCTL[PRDLD] = 1, then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. • The active and shadow registers share the same memory map address.
15-0	TBCTR	R/W	0h	Time-Base Counter Register Reading these bits gives the current time-base counter value. Writing to these bits sets the current time-base counter value. The update happens as soon as the write occurs; the write is NOT synchronized to the time-base clock (TBCLK) and the register is not shadowed.

20.4.4 CMPCTL Register (Offset = Ch) [reset = 0h]

CMPCTL is shown in Figure 20-65 and described in Table 20-27.

Return to Summary Table.

Counter-Compare Control Register

Figure 20-65. CMPCTL Register

31	30	29	28	27	26	25	24
RESERVED						SHDWBFULL	SHDWAFULL
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	SHDWBMODE	RESERVED	SHDWAMODE	LOADBMODE		LOADAMODE	
R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 20-27. CMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	SHDWBFULL	R/W	0h	Counter-compare B (CMPB) Shadow Register Full Status Flag This bit self clears once a load-strobe occurs. 0 CMPB shadow FIFO not full yet 1 Indicates the CMPB shadow FIFO is full; a CPU write will overwrite current shadow value.
24	SHDWAFULL	R/W	0h	Counter-compare A (CMPA) Shadow Register Full Status Flag The flag bit is set when a 32-bit write to CMPA:CMPAHR register or a 16-bit write to CMPA register is made. A 16-bit write to CMPAHR register will not affect the flag. This bit self clears once a load-strobe occurs. 0 CMPA shadow FIFO not full yet 1 Indicates the CMPA shadow FIFO is full, a CPU write will overwrite the current shadow value
23	RESERVED	R	0h	Reserved
22	SHDWBMODE	R/W	0h	Counter-compare B (CMPB) Register Operating Mode 0 Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1 Immediate mode. Only the active compare B register is used. All writes and reads directly access the active register for immediate compare action.
21	RESERVED	R	0h	Reserved
20	SHDWAMODE	R/W	0h	Counter-compare A (CMPA) Register Operating Mode 0 Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1 Immediate mode. Only the active compare register is used. All writes and reads directly access the active register for immediate compare action
19-18	LOADBMODE	R/W	0h	Active Counter-Compare B (CMPB) Load From Shadow Select Mode This bit has no effect in immediate mode (CMPCTL[SHDWBMODE] = 1). 0 Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 1h Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 2h Load on either CTR = Zero or CTR = PRD 3h Freeze (no loads possible)

Table 20-27. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-16	LOADAMODE	R/W	0h	Active Counter-Compare A (CMPA) Load From Shadow Select Mode. This bit has no effect in immediate mode (CMPCTL[SHDWAMODE] = 1). 0 Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 1h Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 2h Load on either CTR = Zero or CTR = PRD 3h Freeze (no loads possible)
15-0	RESERVED	R	0h	Reserved

20.4.5 CMPA Register (Offset = 10h) [reset = 0h]

CMPA is shown in Figure 20-66 and described in Table 20-28.

Return to Summary Table.

Counter-Compare A Register

Figure 20-66. CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA																RESERVED															
R/W-0h																R-0h															

Table 20-28. CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CMPA	R/W	0h	Counter-Compare A Register The value in the active CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare A" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include: • Do nothing; the event is ignored. • Clear: Pull the EPWMxA and/or EPWMxB signal low • Set: Pull the EPWMxA and/or EPWMxB signal high • Toggle the EPWMxA and/or EPWMxB signal Shadowing of this register is enabled and disabled by the CMPCTL[SHDWAMODE] bit. By default this register is shadowed. • If CMPCTL[SHDWAMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADAMODE] bit field determines which event will load the active register from the shadow register. • Before a write, the CMPCTL[SHDWAFULL] bit can be read to determine if the shadow register is currently full. • If CMPCTL[SHDWAMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. • In either mode, the active and shadow registers share the same memory map address.
15-0	RESERVED	R	0h	Reserved

20.4.6 CMPB_AQCTLA Register (Offset = 14h) [reset = 0h]

CMPB_AQCTLA is shown in Figure 20-67 and described in Table 20-29.

Return to Summary Table.

Counter-Compare B Register/ Action-Qualifier Control Register for Output A (EPWMxA)

Figure 20-67. CMPB_AQCTLA Register

31	30	29	28	27	26	25	24
RESERVED				AQCTLA_CBD		AQCTLA_CBU	
R-0h				R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
AQCTLA_CAD		AQCTLA_CAU		AQCTLA_PRD		AQCTLA_ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
CMPB							
R/W-0h							
7	6	5	4	3	2	1	0
CMPB							
R/W-0h							

Table 20-29. CMPB_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-26	AQCTLA_CBD	R/W	0h	Action when the time-base counter equals the active CMPB register and the counter is decrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxA output low. 2h Set: force EPWMxA output high. 3h Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low
25-24	AQCTLA_CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxA output low. 2h Set: force EPWMxA output high. 3h Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low
23-22	AQCTLA_CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxA output low. 2h Set: force EPWMxA output high. 3h Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low
21-20	AQCTLA_CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxA output low. 2h Set: force EPWMxA output high. 3h Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low
19-18	AQCTLA_PRD	R/W	0h	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0 Do nothing (action disabled) 1h Clear: force EPWMxA output low. 2h Set: force EPWMxA output high. 3h Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low
17-16	AQCTLA_ZRO	R/W	0h	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0 Do nothing (action disabled) 1h Clear: force EPWMxA output low. 2h Set: force EPWMxA output high. 3h Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low

Table 20-29. CMPB_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	CMPB	R/W	0h	The value in the active CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare B" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include: • Do nothing. event is ignored. • Clear: Pull the EPWMxA and/or EPWMxB signal low • Set: Pull the EPWMxA and/or EPWMxB signal high • Toggle the EPWMxA and/or EPWMxB signal Shadowing of this register is enabled and disabled by the CMPCTL[SHDWBMODE] bit. By default this register is shadowed. • If CMPCTL[SHDWBMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADBMODE] bit field determines which event will load the active register from the shadow register: • Before a write, the CMPCTL[SHDWBFULL] bit can be read to determine if the shadow register is currently full. • If CMPCTL[SHDWBMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. • In either mode, the active and shadow registers share the same memory map address

20.4.7 AQCTLB_AQSFRC Register (Offset = 18h) [reset = 0h]

AQCTLB_AQSFRC is shown in Figure 20-68 and described in Table 20-30.

Return to Summary Table.

Action-Qualifier Control Register for Output B (EPWMxB)/ Action-Qualifier Software Force Register

Figure 20-68. AQCTLB_AQSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
AQSFRC_RLDCSF		AQSFRC_OTSFB	AQSFRC_ACTSFB		AQSFRC_OTSFA	AQSFRC_ACTSFA	
R/W-0h		R/W-0h	R/W-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED				AQCTLB_CBD		AQCTLB_CBU	
R-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
AQCTLB_CAD		AQCTLB_CAU		AQCTLB_PRD		AQCTLB_ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 20-30. AQCTLB_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-22	AQSFRC_RLDCSF	R/W	0h	AQCSFRC Active Register Reload From Shadow Options 0 Load on event counter equals zero 1h Load on event counter equals period 2h Load on event counter equals zero or counter equals period 3h Load immediately (the active register is directly accessed by the CPU and is not loaded from the shadow register).
21	AQSFRC_OTSFB	R/W	0h	One-Time Software Forced Event on Output B 0 Writing a 0 has no effect. Always reads back a 0 This bit is auto cleared once a write to this register is complete (that is, a forced event is initiated.) This is a one-shot forced event. It can be overridden by another subsequent event on output B. 1 Initiates a single s/w forced event
20-19	AQSFRC_ACTSFB	R/W	0h	Action when One-Time Software Force B Is invoked 0 Does nothing (action disabled) 1h Clear (low) 2h Set (high) 3h Toggle (Low -> High, High -> Low) Note: This action is not qualified by counter direction (CNT_dir)
18	AQSFRC_OTSFA	R/W	0h	One-Time Software Forced Event on Output A 0 Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (that is, a forced event is initiated). 1 Initiates a single software forced event
17-16	AQSFRC_ACTSFA	R/W	0h	Action When One-Time Software Force A Is Invoked 0 Does nothing (action disabled) 1h Clear (low) 2h Set (high) 3h Toggle (Low → High, High → Low) Note: This action is not qualified by counter direction (CNT_dir)
15-12	RESERVED	R	0h	Reserved
11-10	AQCTLB_CBD	R/W	0h	Action when the counter equals the active CMPB register and the counter is decrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxB output low. 2h Set: force EPWMxB output high. 3h Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low
9-8	AQCTLB_CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxB output low. 2h Set: force EPWMxB output high. 3h Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low

Table 20-30. AQCTLB_AQSFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	AQCTLB_CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxB output low. 2h Set: force EPWMxB output high. 3h Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low
5-4	AQCTLB_CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0 Do nothing (action disabled) 1h Clear: force EPWMxB output low. 2h Set: force EPWMxB output high. 3h Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low
3-2	AQCTLB_PRD	R/W	0h	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0 Do nothing (action disabled) 1h Clear: force EPWMxB output low. 2h Set: force EPWMxB output high. 3h Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low
1-0	AQCTLB_ZRO	R/W	0h	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0 Do nothing (action disabled) 1h Clear: force EPWMxB output low. 2h Set: force EPWMxB output high. 3h Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.

20.4.8 AQCSFRC_DBCTL Register (Offset = 1Ch) [reset = 0h]

AQCSFRC_DBCTL is shown in Figure 20-69 and described in Table 20-31.

Return to Summary Table.

Dead-Band Generator Control Register/ Action-Qualifier Continuous S/W Force Register Set

Figure 20-69. AQCSFRC_DBCTL Register

31	30	29	28	27	26	25	24
DBCTL_HALFCYCLE	RESERVED						
R/W-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED		DBCTL_IN_MODE		DBCTL_POLSEL		DBCTL_OUT_MODE	
R-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				AQCSFRC_CSFB		AQCSFRC_CSFA	
R-0h				R/W-0h		R/W-0h	

Table 20-31. AQCSFRC_DBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DBCTL_HALFCYCLE	R/W	0h	Half Cycle Clocking Enable Bit: 0 Full cycle clocking enabled. The dead-band counters are clocked at the TBCLK rate. 1 Half cycle clocking enabled. The dead-band counters are clocked at TBCLK ÷ 2.
30-22	RESERVED	R	0h	Reserved
21-20	DBCTL_IN_MODE	R/W	0h	Dead Band Input Mode Control Bit 5 controls the S5 switch and bit 4 controls the S4 switch shown in Figure 35-28. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms the default is EPWMxA In is the source for both falling and rising-edge delays. 0 EPWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 1h EPWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 2h EPWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 3h EPWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal.
19-18	DBCTL_POLSEL	R/W	0h	Polarity Select Control Bit 3 controls the S3 switch and bit 2 controls the S2 switch shown in Figure 35-28. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0,0. Other enhanced modes are also possible, but not regarded as typical usage modes. 0 Active high (AH) mode. Neither EPWMxA nor EPWMxB is inverted (default). 1h Active low complementary (ALC) mode. EPWMxA is inverted. 2h Active high complementary (AHC). EPWMxB is inverted. 3h Active low (AL) mode. Both EPWMxA and EPWMxB are inverted

Table 20-31. AQCSFRC_DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-16	DBCTL_OUT_MODE	R/W	0h	Dead-band Output Mode Control Bit 1 controls the S1 switch and bit 0 controls the S0 switch shown in Figure 35-28. This allows you to selectively enable or bypass the dead-band generation for the falling-edge and rising-edge delay. 0 Dead-band generation is bypassed for both output signals. In this mode, both the EPWMxA and EPWMxB output signals from the action-qualifier are passed directly to the PWM-chopper submodule. In this mode, the POLSEL and IN_MODE bits have no effect. 1h Disable rising-edge delay. The EPWMxA signal from the action-qualifier is passed straight through to the EPWMxA input of the PWM-chopper submodule. The falling-edge delayed signal is seen on output EPWMxB. The input signal for the delay is determined by DBCTL[IN_MODE]. 2h The rising-edge delayed signal is seen on output EPWMxA. The input signal for the delay is determined by DBCTL[IN_MODE]. Disable falling-edge delay. The EPWMxB signal from the action-qualifier is passed straight through to the EPWMxB input of the PWM-chopper submodule. 3h Dead-band is fully enabled for both rising-edge delay on output EPWMxA and falling-edge delay on output EPWMxB. The input signal for the delay is determined by DBCTL[IN_MODE].
15-4	RESERVED	R	0h	Reserved
3-2	AQCSFRC_CSFB	R/W	0h	Continuous Software Force on Output B In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. To configure shadow mode, use AQSFRC[RLDCSF]. 0 Forcing disabled, that is, has no effect 1h Forces a continuous low on output B 2h Forces a continuous high on output B 3h Software forcing is disabled and has no effect
1-0	AQCSFRC_CSFA	R/W	0h	Continuous Software Force on Output A In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. 0 Forcing disabled, that is, has no effect 1h Forces a continuous low on output A 2h Forces a continuous high on output A 3h Software forcing is disabled and has no effect

20.4.9 DBRED_DBFED Register (Offset = 20h) [reset = 0h]

DBRED_DBFED is shown in Figure 20-70 and described in Table 20-32.

Return to Summary Table.

Dead-Band Generator Rising Edge Delay Count Register/ Dead-Band Generator Falling Edge Delay Count Register

Figure 20-70. DBRED_DBFED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED						DBFED_DEL									
R-0h						R/W-0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						DBRED_DEL									
R-0h						R/W-0h									

Table 20-32. DBRED_DBFED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25-16	DBFED_DEL	R/W	0h	Falling Edge Delay Count. 10-bit counter
15-10	RESERVED	R	0h	Reserved
9-0	DBRED_DEL	R/W	0h	Rising Edge Delay Count. 10-bit counter

20.4.10 TZSEL_TZDCSEL Register (Offset = 24h) [reset = 0h]

TZSEL_TZDCSEL is shown in Figure 20-71 and described in Table 20-33.

Return to Summary Table.

Trip Zone Digital Compare Select Register/ Trip-Zone Select Register

Figure 20-71. TZSEL_TZDCSEL Register

31	30	29	28	27	26	25	24
RESERVED				TZDCSEL_DCBEVT2			TZDCSEL_DCBEVT1
R-0h				R/W-0h			R/W-0h
23	22	21	20	19	18	17	16
TZDCSEL_DCBEVT1		TZDCSEL_DCAEVT2			TZDCSEL_DCAEVT1		
R/W-0h		R/W-0h			R/W-0h		
15	14	13	12	11	10	9	8
TZSEL_DCBEVT1	TZSEL_DCAEVT1	TZSEL_OSHT6	TZSEL_OSHT5	TZSEL_OSHT4	TZSEL_OSHT3	TZSEL_OSHT2	TZSEL_OSHT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TZSEL_DCBEVT2	TZSEL_DCAEVT2	TZSEL_CBC6	TZSEL_CBC5	TZSEL_CBC4	TZSEL_CBC3	TZSEL_CBC2	TZSEL_CBC1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-33. TZSEL_TZDCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-25	TZDCSEL_DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Selection 0 Event disabled 1h DCBH = low, DCBL = don't care 2h DCBH = high, DCBL = don't care 3h DCBL = low, DCBH = don't care 4h DCBL = high, DCBH = don't care 5h DCBL = high, DCBH = low 6h-7h Reserved
24-22	TZDCSEL_DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Selection 0 Event disabled 1h DCBH = low, DCBL = don't care 2h DCBH = high, DCBL = don't care 3h DCBL = low, DCBH = don't care 4h DCBL = high, DCBH = don't care 5h DCBL = high, DCBH = low 6h-7h Reserved
21-19	TZDCSEL_DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Selection 0 Event disabled 1h DCAH = low, DCAL = don't care 2h DCAH = high, DCAL = don't care 3h DCAL = low, DCAH = don't care 4h DCAL = high, DCAH = don't care 5h DCAL = high, DCAH = low 6h-7h Reserved
18-16	TZDCSEL_DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Selection 0 Event disabled 1h DCAH = low, DCAL = don't care 2h DCAH = high, DCAL = don't care 3h DCAL = low, DCAH = don't care 4h DCAL = high, DCAH = don't care 5h DCAL = high, DCAH = low 6h-7h Reserved
15	TZSEL_DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Select 0 Disable DCBEVT1 as one-shot-trip source for this ePWM module. 1 Enable DCBEVT1 as one-shot-trip source for this ePWM module
14	TZSEL_DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Select 0 Disable DCAEVT1 as one-shot-trip source for this ePWM module. 1 Enable DCAEVT1 as one-shot-trip source for this ePWM module
13	TZSEL_OSHT6	R/W	0h	Trip-zone 6 (TZ6) Select 0 Disable TZ6 as a one-shot trip source for this ePWM module. 1 Enable TZ6 as a one-shot trip source for this ePWM module
12	TZSEL_OSHT5	R/W	0h	Trip-zone 5 (TZ5) Select 0 Disable TZ5 as a one-shot trip source for this ePWM module 1 Enable TZ5 as a one-shot trip source for this ePWM module

Table 20-33. TZSEL_TZDCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	TZSEL_OSHT4	R/W	0h	Trip-zone 4 (TZ4) Select 0 Disable TZ4 as a one-shot trip source for this ePWM module 1 Enable TZ4 as a one-shot trip source for this ePWM module
10	TZSEL_OSHT3	R/W	0h	Trip-zone 3 (TZ3) Select 0 Disable TZ3 as a one-shot trip source for this ePWM module 1 Enable TZ3 as a one-shot trip source for this ePWM module
9	TZSEL_OSHT2	R/W	0h	Trip-zone 2 (TZ2) Select 0 Disable TZ2 as a one-shot trip source for this ePWM module 1 Enable TZ2 as a one-shot trip source for this ePWM module
8	TZSEL_OSHT1	R/W	0h	Trip-zone 1 (TZ1) Select 0 Disable TZ1 as a one-shot trip source for this ePWM module 1 Enable TZ1 as a one-shot trip source for this ePWM module
7	TZSEL_DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Select 0 Disable DCBEVT2 as a CBC trip source for this ePWM module 1 Enable DCBEVT2 as a CBC trip source for this ePWM module
6	TZSEL_DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Select 0 Disable DCAEVT2 as a CBC trip source for this ePWM module 1 Enable DCAEVT2 as a CBC trip source for this ePWM module
5	TZSEL_CBC6	R/W	0h	Trip-zone 6 (TZ6) Select 0 Disable TZ6 as a CBC trip source for this ePWM module 1 Enable TZ6 as a CBC trip source for this ePWM module
4	TZSEL_CBC5	R/W	0h	Trip-zone 5 (TZ5) Select 0 Disable TZ5 as a CBC trip source for this ePWM module 1 Enable TZ5 as a CBC trip source for this ePWM module
3	TZSEL_CBC4	R/W	0h	Trip-zone 4 (TZ4) Select 0 Disable TZ4 as a CBC trip source for this ePWM module 1 Enable TZ4 as a CBC trip source for this ePWM module
2	TZSEL_CBC3	R/W	0h	Trip-zone 3 (TZ3) Select 0 Disable TZ3 as a CBC trip source for this ePWM module 1 Enable TZ3 as a CBC trip source for this ePWM module
1	TZSEL_CBC2	R/W	0h	Trip-zone 2 (TZ2) Select 0 Disable TZ2 as a CBC trip source for this ePWM module 1 Enable TZ2 as a CBC trip source for this ePWM module
0	TZSEL_CBC1	R/W	0h	Trip-zone 1 (TZ1) Select 0 Disable TZ1 as a CBC trip source for this ePWM module 1 Enable TZ1 as a CBC trip source for this ePWM module

20.4.11 TZCTL_TZEINT Register (Offset = 28h) [reset = 0h]

TZCTL_TZEINT is shown in Figure 20-72 and described in Table 20-34.

Return to Summary Table.

Trip-Zone Control Register/ Trip-Zone Enable Interrupt Register

Figure 20-72. TZCTL_TZEINT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	TZEINT_DCBEVT2	TZEINT_DCBEVT1	TZEINT_DCAEVT2	TZEINT_DCAEVT1	TZEINT_OST	TZEINT_CBC	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				TZCTL_DCBEVT2		TZCTL_DCBEVT1	
R-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
TZCTL_DCAEVT2		TZCTL_DCAEVT1		TZCTL_TZB		TZCTL_TZA	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 20-34. TZCTL_TZEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22	TZEINT_DCBEVT2	R/W	0h	Digital Comparator Output B Event 2 Interrupt Enable 0 Disabled 1 Enabled
21	TZEINT_DCBEVT1	R/W	0h	Digital Comparator Output B Event 1 Interrupt Enable 0 Disabled 1 Enabled
20	TZEINT_DCAEVT2	R/W	0h	Digital Comparator Output A Event 2 Interrupt Enable 0 Disabled 1 Enabled
19	TZEINT_DCAEVT1	R/W	0h	Digital Comparator Output A Event 1 Interrupt Enable 0 Disabled 1 Enabled
18	TZEINT_OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 0 Disable one-shot interrupt generation 1 Enable Interrupt generation; a one-shot trip event will cause a EPWMx_TZINT VIM interrupt
17	TZEINT_CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 0 Disable cycle-by-cycle interrupt generation. 1 Enable interrupt generation; a cycle-by-cycle trip event will cause an EPWMx_TZINT VIM interrupt
16	RESERVED	R	0h	Reserved
15-12	RESERVED	R	0h	Reserved
11-10	TZCTL_DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Action On EPWMxB: 0 High-impedance (EPWMxB = High-impedance state) 1h Force EPWMxB to a high state. 2h Force EPWMxB to a low state. 3h Do Nothing, trip action is disabled
9-8	TZCTL_DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Action On EPWMxB: 0 High-impedance (EPWMxB = High-impedance state) 1h Force EPWMxB to a high state. 2h Force EPWMxB to a low state. 3h Do Nothing, trip action is disabled
7-6	TZCTL_DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Action On EPWMxA: 0 High-impedance (EPWMxA = High-impedance state) 1h Force EPWMxA to a high state. 2h Force EPWMxA to a low state. 3h Do Nothing, trip action is disabled

Table 20-34. TZCTL_TZEINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	TZCTL_DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Action On EPWMxA: 0 High-impedance (EPWMxA = High-impedance state) 1h Force EPWMxA to a high state. 2h Force EPWMxA to a low state. 3h Do Nothing, trip action is disabled
3-2	TZCTL_TZB	R/W	0h	When a trip event occurs the following action is taken on output EPWMxB. Which trip-zone pins can cause an event is defined in the TZSEL register. 0 High-impedance (EPWMxB = High-impedance state) 1h Force EPWMxB to a high state 2h Force EPWMxB to a low state 3h Do nothing, no action is taken on EPWMxB.
1-0	TZCTL_TZA	R/W	0h	When a trip event occurs the following action is taken on output EPWMxA. Which trip-zone pins can cause an event is defined in the TZSEL register. 0 High-impedance (EPWMxA = High-impedance state) 1h Force EPWMxA to a high state 2h Force EPWMxA to a low state 3h Do nothing, no action is taken on EPWMxA

20.4.12 TZFLG_TZCLR Register (Offset = 2Ch) [reset = 0h]

TZFLG_TZCLR is shown in Figure 20-73 and described in Table 20-35.

Return to Summary Table.

Trip-Zone Flag Register/ Trip-Zone Clear Register

Figure 20-73. TZFLG_TZCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	TZCLR_DCBEVT2	TZCLR_DCBEVT1	TZCLR_DCAEVT2	TZCLR_DCAEVT1	TZCLR_OST	TZCLR_CBC	TZCLR_INT
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	TZFLG_DCBEVT2	TZFLG_DCBEVT1	TZFLG_DCAEVT2	TZFLG_DCAEVT1	TZFLG_OST	TZFLG_CBC	TZFLG_INT
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-35. TZFLG_TZCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	Reserved
22	TZCLR_DCBEVT2	R/W	0h	Clear Flag for Digital Compare Output B Event 2 0 Writing 0 has no effect. This bit always reads back 0. 1 Writing 1 clears the DCBEVT2 event trip condition.
21	TZCLR_DCBEVT1	R/W	0h	Clear Flag for Digital Compare Output B Event 1 0 Writing 0 has no effect. This bit always reads back 0. 1 Writing 1 clears the DCBEVT1 event trip condition
20	TZCLR_DCAEVT2	R/W	0h	Clear Flag for Digital Compare Output A Event 2 0 Writing 0 has no effect. This bit always reads back 0. 1 Writing 1 clears the DCAEVT2 event trip condition
19	TZCLR_DCAEVT1	R/W	0h	Clear Flag for Digital Compare Output A Event 1 0 Writing 0 has no effect. This bit always reads back 0. 1 Writing 1 clears the DCAEVT1 event trip condition.
18	TZCLR_OST	R/W	0h	Clear Flag for One-Shot Trip (OST) Latch 0 Has no effect. Always reads back a 0. 1 Clears this Trip (set) condition.
17	TZCLR_CBC	R/W	0h	Clear Flag for Cycle-By-Cycle (CBC) Trip Latch 0 Has no effect. Always reads back a 0. 1 Clears this Trip (set) condition.
16	TZCLR_INT	R/W	0h	Global Interrupt Clear Flag 0 Has no effect. Always reads back a 0. 1 Clears the trip-interrupt flag for this ePWM module (TZFLG[INT]). NOTE: No further EPWMx_TZINT VIM interrupts will be generated until the flag is cleared. If the TZFLG[INT] bit is cleared and any of the other flag bits are set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts.
15-7	RESERVED	R	0h	Reserved
6	TZFLG_DCBEVT2	R/W	0h	Latched Status Flag for Digital Compare Output B Event 2 0 Indicates no trip event has occurred on DCBEVT2 1 Indicates a trip event has occurred for the event defined for DCBEVT2
5	TZFLG_DCBEVT1	R/W	0h	Latched Status Flag for Digital Compare Output B Event 1 0 Indicates no trip event has occurred on DCBEVT1 1 Indicates a trip event has occurred for the event defined for DCBEVT1

Table 20-35. TZFLG_TZCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TZFLG_DCAEVT2	R/W	0h	Latched Status Flag for Digital Compare Output A Event 2 0 Indicates no trip event has occurred on DCAEVT2 1 Indicates a trip event has occurred for the event defined for DCAEVT2
3	TZFLG_DCAEVT1	R/W	0h	Latched Status Flag for Digital Compare Output A Event 1 0 Indicates no trip event has occurred on DCAEVT1 1 Indicates a trip event has occurred for the event defined for DCAEVT1
2	TZFLG_OST	R/W	0h	Latched Status Flag for A One-Shot Trip Event 0 No one-shot trip event has occurred. 1 Indicates a trip event has occurred on a pin selected as a one-shot trip source. This bit is cleared by writing the appropriate value to the TZCLR register
1	TZFLG_CBC	R/W	0h	Latched Status Flag for Cycle-By-Cycle Trip Event 0 No cycle-by-cycle trip event has occurred. 1 Indicates a trip event has occurred on a signal selected as a cycle-by-cycle trip source. The TZFLG[CBC] bit will remain set until it is manually cleared by the user. If the cycle-by-cycle trip event is still present when the CBC bit is cleared, then CBC will be immediately set again. The specified condition on the signal is automatically cleared when the ePWM time-base counter reaches zero (TBCTR = 0x0000) if the trip condition is no longer present. The condition on the signal is only cleared when the TBCTR = 0x0000 no matter where in the cycle the CBC flag is cleared. This bit is cleared by writing the appropriate value to the TZCLR register
0	TZFLG_INT	R/W	0h	Latched Trip Interrupt Status Flag 0 Indicates no interrupt has been generated. 1 Indicates an EPWMx_TZINT VIM interrupt was generated because of a trip condition. No further EPWMx_TZINT VIM interrupts will be generated until this flag is cleared. If the interrupt flag is cleared when either CBC or OST is set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. This bit is cleared by writing the appropriate value to the TZCLR register

20.4.13 TZFRC_ETSEL Register (Offset = 30h) [reset = 0h]

TZFRC_ETSEL is shown in Figure 20-74 and described in Table 20-36.

Return to Summary Table.

Trip-Zone Force Register / Event-Trigger Selection Register

Figure 20-74. TZFRC_ETSEL Register

31	30	29	28	27	26	25	24
ETSEL_SOCBEN	ETSEL_SOCBSEL			ETSEL_SOCAEN	ETSEL_SOCASEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED				ETSEL_INTEN	ETSEL_INTSEL		
R-0h				R/W-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	TZFRC_DCBEVT2	TZFRC_DCBEVT1	TZFRC_DCAEVT2	TZFRC_DCAEVT1	TZFRC_OST	TZFRC_CBC	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 20-36. TZFRC_ETSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ETSEL_SOCBEN	R/W	0h	Enable the ADC Start of Conversion B (EPWMxSOCB) Pulse 0 Disable EPWMxSOCB. 1 Enable EPWMxSOCB pulse
30-28	ETSEL_SOCBSEL	R/W	0h	EPWMxSOCB Selection Options These bits determine when a EPWMxSOCB pulse will be generated. 0 Enable DCBEVT1.soc event 1h Enable event time-base counter equal to zero. (TBCTR = 0x0000) 2h Enable event time-base counter equal to period (TBCTR = TBPRD) 3h Enable event time-base counter equal to zero or period (TBCTR = 0x0000 or TBCTR = TBPRD). This mode is useful in up-down count mode. 4h Enable event time-base counter equal to CMPA when the timer is incrementing. 5h Enable event time-base counter equal to CMPA when the timer is decrementing. 6h Enable event: time-base counter equal to CMPB when the timer is incrementing. 7h Enable event: time-base counter equal to CMPB when the timer is decrementing
27	ETSEL_SOCAEN	R/W	0h	Enable the ADC Start of Conversion A (EPWMxSOCA) Pulse 0 Disable EPWMxSOCA. 1 Enable EPWMxSOCA pulse.
26-24	ETSEL_SOCASEL	R/W	0h	EPWMxSOCA Selection Options These bits determine when a EPWMxSOCA pulse will be generated. 0 Enable DCAEVT1.soc event 1h Enable event time-base counter equal to zero. (TBCTR = 0x0000) 2h Enable event time-base counter equal to period (TBCTR = TBPRD) 3h Enable event time-base counter equal to zero or period (TBCTR = 0x0000 or TBCTR = TBPRD). This mode is useful in up-down count mode. 4h Enable event time-base counter equal to CMPA when the timer is incrementing. 5h Enable event time-base counter equal to CMPA when the timer is decrementing. 6h Enable event: time-base counter equal to CMPB when the timer is incrementing. 7h Enable event: time-base counter equal to CMPB when the timer is decrementing
23-20	RESERVED	R	0h	Reserved
19	ETSEL_INTEN	R/W	0h	Enable ePWM Interrupt (EPWMx_INT) Generation 0 Disable EPWMx_INT generation 1 Enable EPWMx_INT generation

Table 20-36. TZFRC_ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	ETSEL_INTSEL	R/W	0h	ePWM Interrupt (EPWMx_INT) Selection Options 0 Reserved 1h Enable event time-base counter equal to zero. (TBCTR = 0x0000) 2h Enable event time-base counter equal to period (TBCTR = TBPRD) 3h Enable event time-base counter equal to zero or period (TBCTR = 0x0000 or TBCTR = TBPRD). This mode is useful in up-down count mode. 4h Enable event time-base counter equal to CMPA when the timer is incrementing. 5h Enable event time-base counter equal to CMPA when the timer is decrementing. 6h Enable event: time-base counter equal to CMPB when the timer is incrementing. 7h Enable event: time-base counter equal to CMPB when the timer is decrementing.
15-7	RESERVED	R	0h	Reserved
6	TZFRC_DCBEVT2	R/W	0h	Force Flag for Digital Compare Output B Event 2 0 Writing 0 has no effect. This bit always reads back 0. 1 Writing 1 forces the DCBEVT2 event trip condition and sets the TZFLG[DCBEVT2] bit.
5	TZFRC_DCBEVT1	R/W	0h	Force Flag for Digital Compare Output B Event 1 0 Writing 0 has no effect. This bit always reads back 0. 1 Writing 1 forces the DCBEVT1 event trip condition and sets the TZFLG[DCBEVT1] bit
4	TZFRC_DCAEVT2	R/W	0h	Force Flag for Digital Compare Output A Event 2 0 Writing 0 has no effect. This bit always reads back 0. 1 Writing 1 forces the DCAEVT2 event trip condition and sets the TZFLG[DCAEVT2] bit
3	TZFRC_DCAEVT1	R/W	0h	Force Flag for Digital Compare Output A Event 1 0 Writing 0 has no effect. This bit always reads back 0 1 Writing 1 forces the DCAEVT1 event trip condition and sets the TZFLG[DCAEVT1] bit
2	TZFRC_OST	R/W	0h	Force a One-Shot Trip Event via Software 0 Writing of 0 is ignored. Always reads back a 0. 1 Forces a one-shot trip event and sets the TZFLG[OST] bit
1	TZFRC_CBC	R/W	0h	Force a Cycle-by-Cycle Trip Event via Software 0 Writing of 0 is ignored. Always reads back a 0. 1 Forces a cycle-by-cycle trip event and sets the TZFLG[CBC] bit.
0	RESERVED	R	0h	Reserved

20.4.14 ETPS_ETFLG Register (Offset = 34h) [reset = 0h]

ETPS_ETFLG is shown in Figure 20-75 and described in Table 20-37.

Return to Summary Table.

Event-Trigger Pre-Scale Register/ Event-Trigger Flag Register

Figure 20-75. ETPS_ETFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ETFLG_SOCB	ETFLG_SOCA	RESERVED	ETFLG_INT
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h
15	14	13	12	11	10	9	8
ETPS_SOCBCNT		ETPS_SOCBPRD		ETPS_SOCACNT		ETPS_SOCAPRD	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				ETPS_INTCNT		ETPS_INTPRD	
R-0h				R/W-0h		R/W-0h	

Table 20-37. ETPS_ETFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ETFLG_SOCB	R/W	0h	Latched ePWM ADC Start-of-Conversion B (EPWMxSOCB) Status Flag 0 Indicates no EPWMxSOCB event occurred 1 Indicates that a start of conversion pulse was generated on EPWMxSOCB. The EPWMxSOCB output will continue to be generated even if the flag bit is set.
18	ETFLG_SOCA	R/W	0h	Latched ePWM ADC Start-of-Conversion A (EPWMxSOCA) Status Flag Unlike the ETFLG[INT] flag, the EPWMxSOCA output will continue to pulse even if the flag bit is set. 0 Indicates no event occurred 1 Indicates that a start of conversion pulse was generated on EPWMxSOCA. The EPWMxSOCA output will continue to be generated even if the flag bit is set
17	RESERVED	R	0h	Reserved
16	ETFLG_INT	R/W	0h	Latched ePWM Interrupt (EPWMx_INT) Status Flag 0 Indicates no event occurred 1 Indicates that an ePWMx interrupt (EWPMx_INT) was generated. No further interrupts will be generated until the flag bit is cleared. Up to one interrupt can be pending while the ETFLG[INT] bit is still set. If an interrupt is pending, it will not be generated until after the ETFLG[INT] bit is cleared
15-14	ETPS_SOCBCNT	R/W	0h	ePWM ADC Start-of-Conversion B Event (EPWMxSOCB) Counter Register These bits indicate how many selected ETSEL[SOCBSEL] events have occurred: 0 No events have occurred. 1h 1 event has occurred. 2h 2 events have occurred. 3h 3 events have occurred
13-12	ETPS_SOCBPRD	R/W	0h	ePWM ADC Start-of-Conversion B Event (EPWMxSOCB) Period Select These bits determine how many selected ETSEL[SOCBSEL] events need to occur before an EPWMxSOCB pulse is generated. To be generated, the pulse must be enabled (ETSEL[SOCBEN] = 1). The SOCB pulse will be generated even if the status flag is set from a previous start of conversion (ETFLG[SOCB] = 1). Once the SOCB pulse is generated, the ETPS[SOCBCNT] bits will automatically be cleared. 0 Disable the SOCB event counter. No EPWMxSOCB pulse will be generated 1h Generate the EPWMxSOCB pulse on the first event: ETPS[SOCBCNT] = 0,1 2h Generate the EPWMxSOCB pulse on the second event: ETPS[SOCBCNT] = 1,0 3h Generate the EPWMxSOCB pulse on the third event: ETPS[SOCBCNT] = 1,1

Table 20-37. ETPS_ETFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	ETPS_SOCACNT	R/W	0h	ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Counter Register These bits indicate how many selected ETSEL[SOCASEL] events have occurred: 0 No events have occurred. 1h 1 event has occurred. 2h 2 events have occurred. 3h 3 events have occurred.
9-8	ETPS_SOCAPRD	R/W	0h	ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Period Select These bits determine how many selected ETSEL[SOCASEL] events need to occur before an EPWMxSOCA pulse is generated. To be generated, the pulse must be enabled (ETSEL[SOCAEN] = 1). The SOCA pulse will be generated even if the status flag is set from a previous start of conversion (ETFLG[SOCA] = 1). Once the SOCA pulse is generated, the ETPS[SOCACNT] bits will automatically be cleared. 0 Disable the SOCA event counter. No EPWMxSOCA pulse will be generated 1h Generate the EPWMxSOCA pulse on the first event: ETPS[SOCACNT] = 0,1 2h Generate the EPWMxSOCA pulse on the second event: ETPS[SOCACNT] = 1,0 3h Generate the EPWMxSOCA pulse on the third event: ETPS[SOCACNT] = 1,1
7-4	RESERVED	R	0h	Reserved
3-2	ETPS_INTCNT	R/W	0h	ePWM Interrupt Event (EPWMx_INT) Counter Register These bits indicate how many selected ETSEL[INTSEL] events have occurred. These bits are automatically cleared when an interrupt pulse is generated. If interrupts are disabled, ETSEL[INT] = 0 or the interrupt flag is set, ETFLG[INT] = 1, the counter will stop counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD]. 0 No events have occurred. 1h 1 event has occurred. 2h 2 events have occurred. 3h 3 events have occurred.
1-0	ETPS_INTPRD	R/W	0h	ePWM Interrupt (EPWMx_INT) Period Select These bits determine how many selected ETSEL[INTSEL] events need to occur before an interrupt is generated. To be generated, the interrupt must be enabled (ETSEL[INT] = 1). If the interrupt status flag is set from a previous interrupt (ETFLG[INT] = 1) then no interrupt will be generated until the flag is cleared via the ETCLR[INT] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETPS[INTCNT] bits will automatically be cleared. Writing a INTPRD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a INTPRD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero INTPRD value is written, the counter is incremented. 0 Disable the interrupt event counter. No interrupt will be generated and ETFRC[INT] is ignored. 1h Generate an interrupt on the first event INTCNT = 01 (first event) 2h Generate interrupt on ETPS[INTCNT] = 1,0 (second event) 3h Generate interrupt on ETPS[INTCNT] = 1,1 (third event)

20.4.15 ETCLR_ETFRC Register (Offset = 38h) [reset = 0h]

ETCLR_ETFRC is shown in Figure 20-76 and described in Table 20-38.

Return to Summary Table.

Event-Trigger Clear Register/ Event-Trigger Force Register

Figure 20-76. ETCLR_ETFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ETFRC_SOCB	ETFRC_SOCA	RESERVED	ETFRC_INT
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ETCLR_SOCB	ETCLR_SOCA	RESERVED	ETCLR_INT
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h

Table 20-38. ETCLR_ETFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ETFRC_SOCB	R/W	0h	SOCB Force Bit. The SOCB pulse will only be generated if the event is enabled in the ETSEL register. The ETFLG[SOCB] flag bit will be set regardless. 0 Has no effect. Always reads back a 0. 1 Generates a pulse on EPWMxSOCB and sets the SOCBFLG bit. This bit is used for test purposes
18	ETFRC_SOCA	R/W	0h	SOCA Force Bit. The SOCA pulse will only be generated if the event is enabled in the ETSEL register. The ETFLG[SOCA] flag bit will be set regardless. 0 Writing 0 to this bit will be ignored. Always reads back a 0. 1 Generates a pulse on EPWMxSOCA and set the SOCAFLG bit. This bit is used for test purposes
17	RESERVED	R	0h	Reserved
16	ETFRC_INT	R/W	0h	INT Force Bit. The interrupt will only be generated if the event is enabled in the ETSEL register. The INT flag bit will be set regardless. 0 Writing 0 to this bit will be ignored. Always reads back a 0. 1 Generates an interrupt on EPWMxINT and set the INT flag bit. This bit is used for test purposes
15-4	RESERVED	R	0h	Reserved
3	ETCLR_SOCB	R/W	0h	ePWM ADC Start-of-Conversion B (EPWMxSOCB) Flag Clear Bit 0 Writing a 0 has no effect. Always reads back a 0 1 Clears the ETFLG[SOCB] flag bit
2	ETCLR_SOCA	R/W	0h	ePWM ADC Start-of-Conversion A (EPWMxSOCA) Flag Clear Bit 0 Writing a 0 has no effect. Always reads back a 0 1 Clears the ETFLG[SOCA] flag bit
1	RESERVED	R	0h	Reserved
0	ETCLR_INT	R/W	0h	ePWM Interrupt (EPWMx_INT) Flag Clear Bit 0 Writing a 0 has no effect. Always reads back a 0 1 Clears the ETFLG[INT] flag bit and enable further interrupts pulses to be generated

20.4.16 PCCTL Register (Offset = 3Ch) [reset = 0h]

PCCTL is shown in Figure 20-77 and described in Table 20-39.

Return to Summary Table.

PWM-Chopper Control Register

Figure 20-77. PCCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					CHPDUTY		
R-0h					R/W-0h		
7	6	5	4	3	2	1	0
CHPFREQ			OSHTWTH				CHPEN
R/W-0h			R/W-0h				R/W-0h

Table 20-39. PCCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10-8	CHPDUTY	R/W	0h	Chopping Clock Duty Cycle 0 Duty = 1/8 (12.5%) 1h Duty = 2/8 (25.0%) 2h Duty = 3/8 (37.5%) 3h Duty = 4/8 (50.0%) 4h Duty = 5/8 (62.5%) 5h Duty = 6/8 (75.0%) 6h Duty = 7/8 (87.5%) 7h Reserved
7-5	CHPFREQ	R/W	0h	Chopping Clock Frequency 0 Divide by 1 (no prescale, = 12.5 MHz at 100 MHz VCLK3) 1h Divide by 2 (6.25 MHz at 100 MHz VCLK3) 2h Divide by 3 (4.16 MHz at 100 MHz VCLK3) 3h Divide by 4 (3.12 MHz at 100 MHz VCLK3) 4h Divide by 5 (2.50 MHz at 100 MHz VCLK3) 5h Divide by 6 (2.08 MHz at 100 MHz VCLK3) 6h Divide by 7 (1.78 MHz at 100 MHz VCLK3) 7h Divide by 8 (1.56 MHz at 100 MHz VCLK3)
4-1	OSHTWTH	R/W	0h	One-Shot Pulse Width 0 1 x VCLK3 / 8 wide (= 80 nS at 100 MHz VCLK3) 1h 2 x VCLK3 / 8 wide (= 160 nS at 100 MHz VCLK3) 2h 3 x VCLK3 / 8 wide (= 240 nS at 100 MHz VCLK3) 3h 4 x VCLK3 / 8 wide (= 320 nS at 100 MHz VCLK3) 4h 5 x VCLK3 / 8 wide (= 400 nS at 100 MHz VCLK3) 5h 6 x VCLK3 / 8 wide (= 480 nS at 100 MHz VCLK3) 6h 7 x VCLK3 / 8 wide (= 560 nS at 100 MHz VCLK3) 7h 8 x VCLK3 / 8 wide (= 640 nS at 100 MHz VCLK3) 8h 9 x VCLK3 / 8 wide (= 720 nS at 100 MHz VCLK3) 9h 10 x VCLK3 / 8 wide (= 800 nS at 100 MHz VCLK3) Ah 11 x VCLK3 / 8 wide (= 880 nS at 100 MHz VCLK3) Bh 12 x VCLK3 / 8 wide (= 960 nS at 100 MHz VCLK3) Ch 13 x VCLK3 / 8 wide (= 1040 nS at 100 MHz VCLK3) Dh 14 x VCLK3 / 8 wide (= 1120 nS at 100 MHz VCLK3) Eh 15 x VCLK3 / 8 wide (= 1200 nS at 100 MHz VCLK3) Fh 16 x VCLK3 / 8 wide (= 1280 nS at 100 MHz VCLK3)
0	CHPEN	R/W	0h	PWM-chopping Enable 0 Disable (bypass) PWM chopping function 1 Enable chopping function

20.4.17 Reserved1 Register (Offset = 40h) [reset = 0h]

Reserved1 is shown in Figure 20-78 and described in Table 20-40.

Return to Summary Table.

Reserved

Figure 20-78. Reserved1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-40. Reserved1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.18 Reserved2 Register (Offset = 44h) [reset = 0h]

Reserved2 is shown in Figure 20-79 and described in Table 20-41.

Return to Summary Table.

Reserved

Figure 20-79. Reserved2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-41. Reserved2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.19 Reserved3 Register (Offset = 48h) [reset = 0h]

Reserved3 is shown in Figure 20-80 and described in Table 20-42.

Return to Summary Table.

Reserved

Figure 20-80. Reserved3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-42. Reserved3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.20 Reserved4 Register (Offset = 4Ch) [reset = 0h]

Reserved4 is shown in Figure 20-81 and described in Table 20-43.

Return to Summary Table.

Reserved

Figure 20-81. Reserved4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-43. Reserved4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.21 Reserved5 Register (Offset = 50h) [reset = 0h]

Reserved5 is shown in Figure 20-82 and described in Table 20-44.

Return to Summary Table.

Reserved

Figure 20-82. Reserved5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-44. Reserved5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.22 Reserved6 Register (Offset = 54h) [reset = 0h]

Reserved6 is shown in Figure 20-83 and described in Table 20-45.

Return to Summary Table.

Reserved

Figure 20-83. Reserved6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-45. Reserved6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.23 Reserved7 Register (Offset = 58h) [reset = 0h]

Reserved7 is shown in Figure 20-84 and described in Table 20-46.

Return to Summary Table.

Reserved

Figure 20-84. Reserved7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-46. Reserved7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.24 Reserved8 Register (Offset = 5Ch) [reset = 0h]

Reserved8 is shown in Figure 20-85 and described in Table 20-47.

Return to Summary Table.

Reserved

Figure 20-85. Reserved8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-0h																															

Table 20-47. Reserved8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

20.4.25 DCTRIPSEL_DCACTL Register (Offset = 60h) [reset = 0h]

DCTRIPSEL_DCACTL is shown in Figure 20-86 and described in Table 20-48.

Return to Summary Table.

Digital Compare Trip Select Register/ Digital Compare A Control Register

Figure 20-86. DCTRIPSEL_DCACTL Register

31	30	29	28	27	26	25	24
RESERVED						DCACTL_EVT2FRC_SYNCSEL	DCACTL_EVT2SRCSEL
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				DCACTL_EVT1SYNCE	DCACTL_EVT1SOCE	DCACTL_EVT1FRC_SYNCSEL	DCACTL_EVT1SRCSEL
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DCTRIPSEL_DCBLCOMPSEL				DCTRIPSEL_DCBHCOMPSEL			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
DCTRIPSEL_DCALCOMPSEL				DCTRIPSEL_DCAHCOMPSEL			
R/W-0h				R/W-0h			

Table 20-48. DCTRIPSEL_DCACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	DCACTL_EVT2FRC_SYNCSEL	R/W	0h	DCAEVT2 Force Synchronization Signal Select 0 Source Is Synchronous Signal 1 Source Is Asynchronous Signal
24	DCACTL_EVT2SRCSEL	R/W	0h	DCAEVT2 Source Signal Select 0 Source Is DCAEVT2 Signal 1 Source Is DCEVTFILT Signal
23-20	RESERVED	R	0h	Reserved
19	DCACTL_EVT1SYNCE	R/W	0h	DCAEVT1 SYNC, Enable/Disable 0 SYNC Generation Disabled 1 SYNC Generation Enabled
18	DCACTL_EVT1SOCE	R/W	0h	DCAEVT1 SOC, Enable/Disable 0 SOC Generation Disabled 1 SOC Generation Enabled
17	DCACTL_EVT1FRC_SYNCSEL	R/W	0h	DCAEVT1 Force Synchronization Signal Select 0 Source Is Synchronous Signal 1 Source Is Asynchronous Signal
16	DCACTL_EVT1SRCSEL	R/W	0h	DCAEVT1 Source Signal Select 0 Source Is DCAEVT1 Signal 1 Source Is DCEVTFILT Signal
15-12	DCTRIPSEL_DCBLCOMPSEL	R/W	0h	Digital Compare B Low Input Select Defines the source for the DCBL input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. 0 TZ1 input 1h TZ2 input 2h TZ3 input All other values Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved.
11-8	DCTRIPSEL_DCBHCOMPSEL	R/W	0h	Digital Compare B High Input Select Defines the source for the DCBH input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. 0 TZ1 input 1h TZ2 input 2h TZ3 input All other values Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved.

Table 20-48. DCTRIPSEL_DCACTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-4	DCTRIPSEL_DCALCOMPSEL	R/W	0h	Digital Compare A Low Input Select Defines the source for the DCAL input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. 0 TZ1 input 1h TZ2 input 2h TZ3 input All other values Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved.
3-0	DCTRIPSEL_DCAHCOMPSEL	R/W	0h	Digital Compare A High Input Select Defines the source for the DCAH input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. 0 TZ1 input 1h TZ2 input 2h TZ3 input All other values Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved.

20.4.26 DCBCTL_DCFCTL Register (Offset = 64h) [reset = 0h]

DCBCTL_DCFCTL is shown in Figure 20-87 and described in Table 20-49.

Return to Summary Table.

Digital Compare B Control Register/ Digital Compare Filter Control Register

Figure 20-87. DCBCTL_DCFCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		DCFCTL_PULSESEL		DCFCTL_BLANKINV	DCFCTL_BLANKE	DCFCTL_SRCSEL	
R-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED						DCBCTL_EVT2FRC_SYNCSEL	DCBCTL_EVT2SRCSEL
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED				DCBCTL_EVT1SYNCE	DCBCTL_EVT1SOCE	DCBCTL_EVT1FRC_SYNCSEL	DCBCTL_EVT1SRCSEL
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-49. DCBCTL_DCFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21-20	DCFCTL_PULSESEL	R/W	0h	Pulse Select For Blanking & Capture Alignment 0 Time-base counter equal to period (TBCTR = TBPRD) 1h Time-base counter equal to zero (TBCTR = 0x0000) 2h-3h Reserved
19	DCFCTL_BLANKINV	R/W	0h	Blanking Window Inversion 0 Blanking window not inverted 1 Blanking window inverted
18	DCFCTL_BLANKE	R/W	0h	Blanking Window Enable/Disable 0 Blanking window is disabled 1 Blanking window is enabled
17-16	DCFCTL_SRCSEL	R/W	0h	Filter Block Signal Source Select 0 Source Is DCAEVT1 Signal 1h Source Is DCAEVT2 Signal 2h Source Is DCBEVT1 Signal 3h Source Is DCBEVT2 Signal
15-10	RESERVED	R	0h	Reserved
9	DCBCTL_EVT2FRC_SYNCSEL	R/W	0h	DCBEVT2 Force Synchronization Signal Select 0 Source Is Synchronous Signal 1 Source Is Asynchronous Signal
8	DCBCTL_EVT2SRCSEL	R/W	0h	DCBEVT2 Source Signal Select 0 Source Is DCBEVT2 Signal 1 Source Is DCEVTFILT Signal
7-4	RESERVED	R	0h	Reserved
3	DCBCTL_EVT1SYNCE	R/W	0h	DCBEVT1 SYNC, Enable/Disable 0 SYNC Generation Disabled 1 SYNC Generation Enabled
2	DCBCTL_EVT1SOCE	R/W	0h	DCBEVT1 SOC, Enable/Disable 0 SOC Generation Disabled 1 SOC Generation Enabled
1	DCBCTL_EVT1FRC_SYNCSEL	R/W	0h	DCBEVT1 Force Synchronization Signal Select 0 Source Is Synchronous Signal 1 Source Is Asynchronous Signal
0	DCBCTL_EVT1SRCSEL	R/W	0h	DCBEVT1 Source Signal Select 0 Source Is DCBEVT1 Signal 1 Source Is DCEVTFILT Signal

20.4.27 DCCAPCTL_DCFOFFSET Register (Offset = 68h) [reset = 0h]

DCCAPCTL_DCFOFFSET is shown in Figure 20-88 and described in Table 20-50.

Return to Summary Table.

Digital Compare Capture Control Register/ Digital Compare Filter Offset Register

Figure 20-88. DCCAPCTL_DCFOFFSET Register

31	30	29	28	27	26	25	24
DCFOFFSET_OFFSET							
R/W-0h							
23	22	21	20	19	18	17	16
DCFOFFSET_OFFSET							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						DCCAPCTL_SHDWMODE	DCCAPCTL_CAPE
R-0h						R/W-0h	R/W-0h

Table 20-50. DCCAPCTL_DCFOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DCFOFFSET_OFFSET	R/W	0h	Blanking Window Offset These 16-bits specify the number of TBCLK cycles from the blanking window reference to the point when the blanking window is applied. The blanking window reference is either period or zero as defined by the DCFCTL[PULSESEL] bit. This offset register is shadowed and the active register is loaded at the reference point defined by DCFCTL[PULSESEL]. The offset counter is also initialized and begins to count down when the active register is loaded. When the counter expires, the blanking window is applied. If the blanking window is currently active, then the blanking window counter is restarted.
15-2	RESERVED	R	0h	Reserved
1	DCCAPCTL_SHDWMODE	R/W	0h	TBCTR Counter Capture Shadow Select Mode 0 Enable shadow mode. The DCCAP active register is copied to shadow register on a TBCTR = TBPRD or TBCTR = zero event as defined by the DCFCTL[PULSESEL] bit. CPU reads of the DCCAP register will return the shadow register contents. 1 Active Mode. In this mode the shadow register is disabled. CPU reads from the DCCAP register will always return the active register contents
0	DCCAPCTL_CAPE	R/W	0h	TBCTR Counter Capture Enable/Disable 0 Disable the time-base counter capture. 1 Enable the time-base counter capture.

20.4.28 DCFOFFSETCNT_DCFWINDOW Register (Offset = 6Ch) [reset = 0h]

DCFOFFSETCNT_DCFWINDOW is shown in Figure 20-89 and described in Table 20-51.

Return to Summary Table.

Digital Compare Filter Offset Counter Register/ Digital Compare Filter Window Register

Figure 20-89. DCFOFFSETCNT_DCFWINDOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								DCFWINDOW_WINDOW							
R-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCFOFFSETCNT_OFFSETCNT															
R/W-0h															

Table 20-51. DCFOFFSETCNT_DCFWINDOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	DCFWINDOW_WINDOW	R/W	0h	Blanking Window Width 0 No blanking window is generated. 1h-FFh Specifies the width of the blanking window in TBCLK cycles. The blanking window begins when the offset counter expires. When this occurs, the window counter is loaded and begins to count down. If the blanking window is currently active and the offset counter expires, the blanking window counter is restarted. The blanking window can cross a PWM period boundary.
15-0	DCFOFFSETCNT_OFFSETCNT	R/W	0h	Blanking Offset Counter These 16-bits are read only and indicate the current value of the offset counter. The counter counts down to zero and then stops until it is re-loaded on the next period or zero event as defined by the DCFCTL[PULSESEL] bit. The offset counter is not affected by the free/soft emulation bits. That is, it will always continue to count down if the device is halted by a emulation stop.

20.4.29 DCFWINDOWCNT_DCCAP Register (Offset = 70h) [reset = 0h]

DCFWINDOWCNT_DCCAP is shown in Figure 20-90 and described in Table 20-52.

Return to Summary Table.

Digital Compare Filter Window Counter Register/ Digital Compare Counter Capture Register

Figure 20-90. DCFWINDOWCNT_DCCAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCCAP															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DCFWINDOWCNT							
R-0h								R/W-0h							

Table 20-52. DCFWINDOWCNT_DCCAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DCCAP	R/W	0h	Digital Compare Time-Base Counter Capture To enable time-base counter capture, set the DCCAPCLT[CAPE] bit to 1. If enabled, reflects the value of the time-base counter (TBCTR) on the low to high edge transition of a filtered (DCEVTFLT) event. Further capture events are ignored until the next period or zero as selected by the DCFCTL[PULSESEL] bit. Shadowing of DCCAP is enabled and disabled by the DCCAPCTL[SHDWMODE] bit. By default this register is shadowed. • If DCCAPCTL[SHDWMODE] = 0, then the shadow is enabled. In this mode, the active register is copied to the shadow register on the TBCTR = TBPRD or TBCTR = zero as defined by the DCFCTL[PULSESEL] bit. CPU reads of this register will return the shadow register value. • If DCCAPCTL[SHDWMODE] = 1, then the shadow register is disabled. In this mode, CPU reads will return the active register value. The active and shadow registers share the same memory map address.
15-8	RESERVED	R	0h	Reserved
7-0	DCFWINDOWCNT	R/W	0h	0-FFh Blanking Window Counter These 8 bits are read only and indicate the current value of the window counter. The counter counts down to zero and then stops until it is re-loaded when the offset counter reaches zero again.

Controller Area Network (DCAN)

This section describes the controller area network (DCAN) module.

NOTE: This section describes a superset implementation of the DCAN module. Consult your device-specific datasheet to identify the applicability of the DMA-related features, the number of instantiations of the DCAN IP, and the number of mailboxes supported on your specific device being used.

Topic **Page**

21.1 Overview 2424
21.2 CAN Blocks 2425
21.3 CAN Bit Timing 2427
21.4 CAN Module Configuration 2431
21.5 Message RAM 2434
21.6 Message Interface Register Sets 2439
21.7 Message Object Configurations 2442
21.8 Message Handling 2444
21.9 CAN Message Transfer 2449
21.10 Interrupt Functionality 2450
21.11 Global Power Down Mode 2452
21.12 Local Power Down Mode 2453
21.13 GIO Support 2453
21.14 Test Modes 2455
21.15 SECDED Mechanism 2459
21.16 Debug/Suspend Mode 2460
21.17 MSS_DCAN Registers 2461

21.1 Overview

The Controller Area Network is a high-integrity, serial, multi-master communication protocol for distributed real-time applications. This CAN module is implemented according to ISO 11898-1 and is suitable for industrial, automotive and general embedded communications.

21.1.1 Features

The DCAN module provides the following features:

Protocol

- Supports CAN protocol version 2.0 part A, B

Speed

- Bit rates up to 1 MBit/s

MailBox

- Configurable Message objects
- Individual identifier masks for each message object
- Programmable FIFO mode for message objects

High Speed MailBox Access

- DMA access to Message RAM.

Power

- Global power down and wakeup support
- Local power down and wakeup support

Debug

- Suspend mode for debug support
- Programmable loop-back modes for self-test operation
- Direct access to Message RAM in test mode
- Supports Two interrupt lines - Level 0 and Level 1

Others

- Automatic Message RAM initialization
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- CAN Rx / Tx pins configurable as general purpose IO pins
- Software module reset
- Message RAM with SECDED mechanism
- Dual clock source to reduce jitter

21.1.2 Functional Description

The CAN protocol is an ISO standard (ISO 11898) for serial data communication. This protocol uses Non-Return To Zero (NRZ) with bit-stuffing. And the communication is carried over a two-wire balanced signaling scheme.

The DCAN data communication happens through the CAN_TX and CAN_RX pins. An additional transceiver hardware is required for the connection to the physical layer (CAN bus) CAN_High and CAN_Low.

The DCAN register set can be accessed directly by the CPU. These registers are used to control and configure the CAN module and the Message RAM.

Individual CAN message objects should be configured for communication over a CAN network. The message objects and identifier masks are stored in the Message RAM.

The CAN module internally handles functions such acceptance filtering, transfer of messages from and to the Message RAM, handling of transmission requests as well as the generation of interrupts or DMA requests.

21.2 CAN Blocks

The DCAN Module, shown in Figure 21-1, comprises of the following basic blocks.

21.2.1 CAN Core

The CAN Core consists of the CAN Protocol Controller and the Rx/Tx Shift Register. It handles all ISO 11898-1 protocol functions.

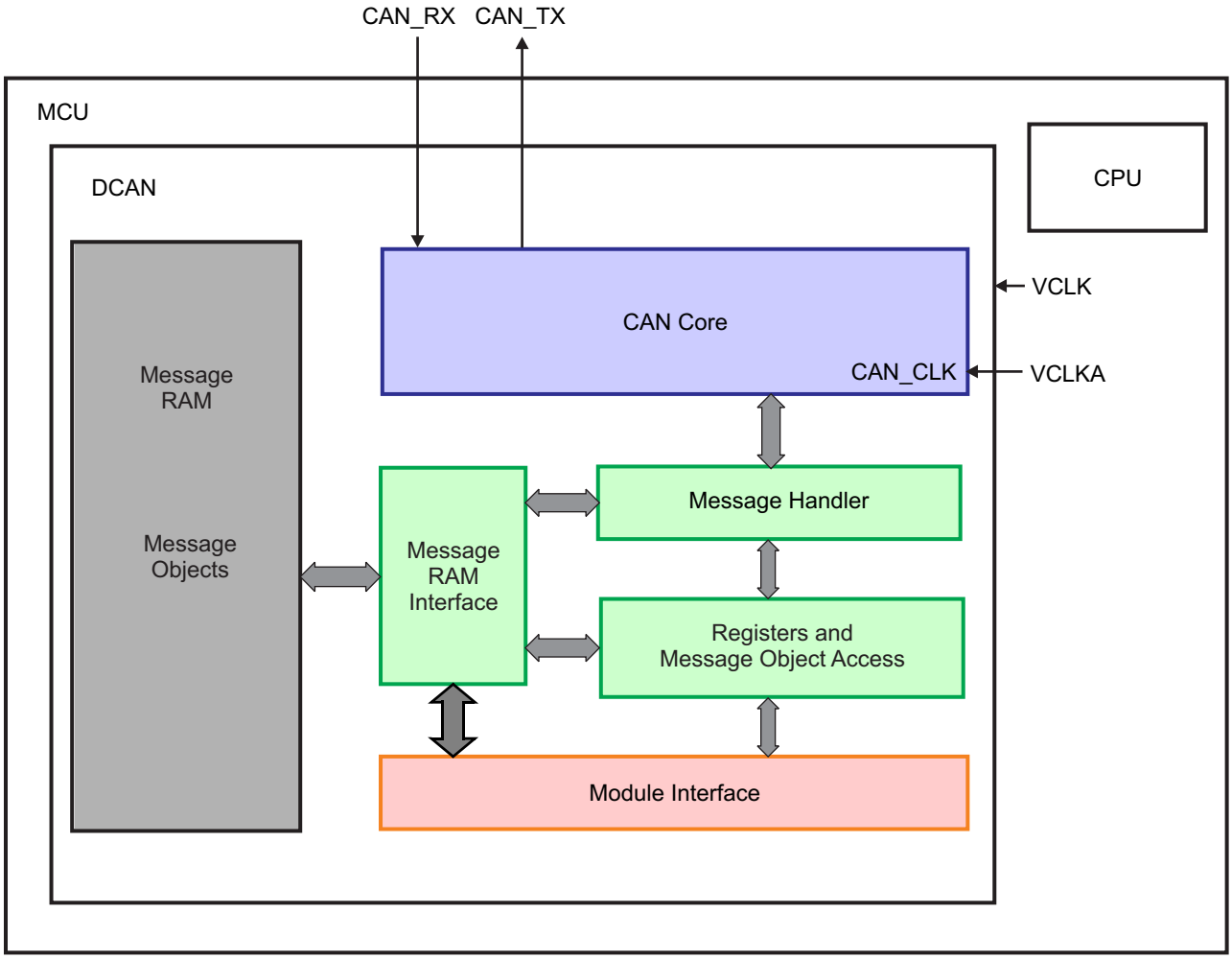
21.2.2 Message RAM

The DCAN Message RAM enables storage of CAN messages. Actual Device datasheet provides the details of the Message RAM address.

21.2.3 Message Handler

The Message Handler is a state machine that controls the data transfer between the single ported Message RAM and the CAN Core's Rx/Tx Shift Register. It also handles acceptance filtering and the interrupt/DMA request generation as programmed in the control registers.

Figure 21-1. DCAN Block Diagram



21.2.4 Message RAM Interface

The Interface Register sets control the CPU read and write accesses to the Message RAM. There are three interface registers IF1, IF2, and IF3:

- IF1 and IF2 Interface Registers sets for read and write access.
- IF3 Interface Register set for read access only.

The Interface Registers have the same word-length as the Message RAM. Additional information can be found in Section 21.6.

21.2.5 Register and Message Object Access

During normal operation, data consistency of the message objects is guaranteed by indirectly accessing the message objects through the interface registers IF1 and IF2.

In order to be able to perform tests on the message object memory, a dedicated test mode has been implemented, that allows direct access by either the CPU or DMA. During normal operation direct access has to be avoided.

21.2.6 Dual Clock Source

Two clock domains are provided to the DCAN module:

1. VCLK - The peripheral synchronous clock domain as the general module clock source.
2. VCLKA - The peripheral asynchronous clock source domain provided to the CAN core as clock source (CAN_CLK) for generating the CAN Bit Timing.

If a frequency modulated clock output from FMPLL is used as the VCLK source, then VCLKA should be derived from an unmodulated clock source (for example, OSCIN source).

The clock source for VCLKA is selected by the Peripheral Asynchronous Clock Source Register in the system module.

Both clock domains can be derived from the same clock source (so that VCLK = VCLKA). However, if frequency modulation in the FMPLL is enabled (spread spectrum clock), then due to the high precision clocking requirements of the CAN Core, the FMPLL clock source should not be used for VCLKA. Alternatively, a separate clock without any modulation (for example, derived directly from the OSCIN clock) should be used for VCLKA.

Refer to the system module reference guide and the device datasheet for more information how to configure the relevant clock source registers in the system module.

Between the two clock domains, a synchronization mechanism is implemented in the DCAN module in order to ensure correct data transfer.

NOTE: If the dual clock functionality is used, then VCLK must always be higher or equal to CAN_CLK (derived from the asynchronous clock source), in order to achieve a stable functionality of the DCAN. Here also the frequency shift of the modulated VCLK has to be considered:

$$f_{0,\ VCLK} \pm \Delta f_{FM,VCLK} \geq f_{CANCLK}$$

The CAN Core has to be programmed to at least 8 clock cycles per bit time. To achieve a transfer rate of 1 MBaud when using the asynchronous clock domain as the clock source for CAN_CLK, an oscillator frequency of 8MHz or higher has to be used.

21.3 CAN Bit Timing

The DCAN supports bit rates between less than 1 kBit/s and 1000 kBit/s.

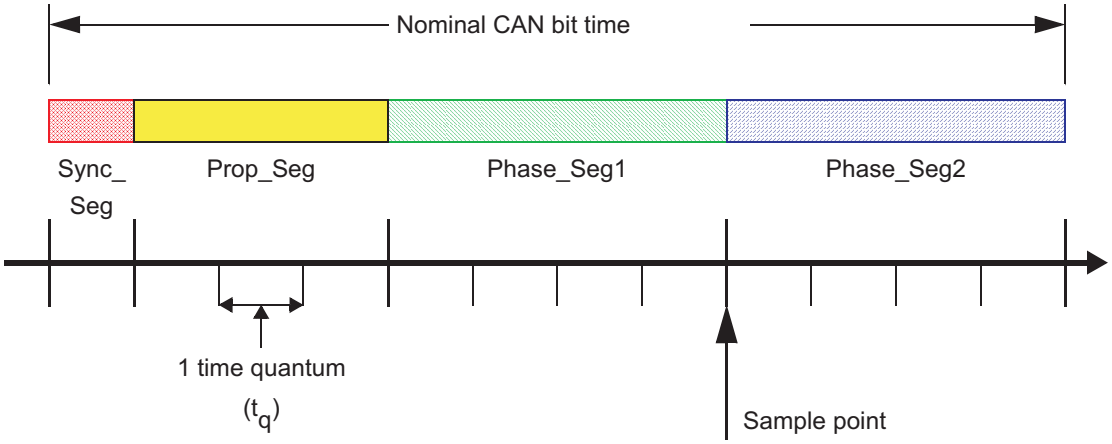
Each member of the CAN network has its own clock generator, typically derived from a crystal oscillator. The Bit timing parameters can be configured individually for each CAN node, creating a common Bit rate even though the CAN nodes' oscillator periods (f_{osc}) may be different.

21.3.1 Bit Time and Bit Rate

According to the CAN specification, the Bit time is divided into four segments (see Figure 21-2):

- Synchronization Segment (Sync_Seg)
- Propagation Time Segment (Prop_Seg)
- Phase Buffer Segment 1 (Phase_Seg1)
- Phase Buffer Segment 2 (Phase_Seg2)

Figure 21-2. Bit Timing



Each segment consists of a specific number of time quanta. The length of one time quantum, (t_q), which is the basic time unit of the bit time, is given by the CAN_CLK and the Baud Rate Prescalers (BRPE and BRP). With these two Baud Rate Prescalers combined, divider values from 1 to 1024 can be programmed:

t_q = Baud Rate Prescaler / CAN_CLK

Apart from the fixed length of the synchronization segment, these numbers are programmable.

Table 21-1 describes the minimum programmable ranges required by the CAN protocol. A given bit rate may be met by different Bit time configurations.

NOTE: For proper functionality of the CAN network, the physical delay times and the oscillator's tolerance range have to be considered.

Table 21-1. Parameters of the CAN Bit Time

Parameter	Range	Remark
Sync_Seg	1 t_q (fixed)	Synchronization of bus input to CAN_CLK
Prop_Seg	[1 … 8] t_q	Compensates for the physical delay times
Phase_Seg1	[1 … 8] t_q	May be lengthened temporarily by synchronization
Phase_Seg2	[1 … 8] t_q	May be shortened temporarily by synchronization
Synchronization Jump Width (SJW)	[1 … 4] t_q	May not be longer than either Phase Buffer Segment

21.3.1.1 Synchronization Segment

The Synchronization Segment (Sync_Seg) is the part of the bit time where edges of the CAN bus level are expected to occur. If an edge occurs outside of Sync_Seg, its distance to the Sync_Seg is called the phase error of this edge.

21.3.1.2 Propagation Time Segment

This part of the bit time is used to compensate physical delay times within the CAN network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

21.3.1.3 Phase Buffer Segments and Synchronization

The Phase Buffer Segments (Phase_Seg1 and Phase_Seg2) and the Synchronization Jump Width (SJW) are used to compensate for the oscillator tolerance.

The Phase Buffer Segments surround the sample point. The Phase Buffer Segments may be lengthened or shortened by synchronization.

The Synchronization Jump Width (SJW) defines how far the resynchronizing mechanism may move the sample point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

Synchronizations occur on edges from recessive to dominant. Their purpose is to control the distance between edges and sample points.

Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous sample point. A synchronization may be done only if a recessive bit was sampled at the previous sample point and if the actual time quantum's bus level is dominant.

An edge is synchronous if it occurs inside of Sync_Seg, otherwise its distance to the Sync_Seg is the edge phase error, measured in time quanta. If the edge occurs before Sync_Seg, the phase error is negative, else it is positive.

21.3.1.4 Oscillator Tolerance Range

With the introduction of CAN protocol version 1.2, the option to synchronize on edges from dominant to recessive became obsolete. Only edges from recessive to dominant are considered for synchronization. The protocol update to version 2.0 (A and B) had no influence on the oscillator tolerance.

The tolerance range df for an oscillator's frequency f_{osc} around the nominal frequency f_{nom} with:

$$(1-df) \times f_{nom} \leq f_{osc} \leq (1+df) \times f_{nom} \quad (5)$$

depends on the proportions of Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df is the defined by two conditions (both shall be met):

$$\text{I:} \quad df \leq \frac{\min(\text{Phase\_Seg1}, \text{Phase\_Seg2})}{[2 \times (13 \times \text{bit\_time} - \text{Phase\_Seg2})]}$$

$$\text{II:} \quad df \leq \frac{\text{SJW}}{20 \times \text{bit\_time}} \quad (6)$$

It has to be considered that SJW may not be larger than the smaller of the Phase Buffer Segments and that the Propagation Time Segment limits that part of the bit time that may be used for the Phase Buffer Segments.

The combination Prop_Seg = 1 and Phase_Seg1 = Phase_Seg2 = SJW = 4 allows the largest possible oscillator tolerance of 1.58%. This combination with a Propagation Time Segment of only 10% of the bit time is not suitable for short bit times; it can be used for bit rates of up to 125 kBit/s (bit time = 8 μs) with a bus length of 40 m.

21.3.2 DCAN Bit Timing Registers

In the DCAN, the bit timing configuration is programmed in two register bytes, additionally a third byte for a baud rate prescaler extension of 4 bits (BREP) is provided. The sum of Prop_Seg and Phase_Seg1 (as TSEG1) is combined with Phase_Seg2 (as TSEG2) in one byte, SJW and BRP (plus BRPE in third byte) are combined in the other byte

In this bit timing register, the components TSEG1, TSEG2, SJW and BRP have to be programmed to a numerical value that is one less than its functional value; so instead of values in the range of [1… n], values in the range of [0 … n-1] are programmed. That way, SJW (functional range of [1 … 4]) is represented by only two bits.

Therefore the length of the Bit time is (programmed values) [TSEG1 + TSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The data in the Bit Timing Register (BTR) is the configuration input of the CAN protocol controller. The Baud Rate Prescaler (configured by BRPE/BRP) defines the length of the time quantum (the basic time unit of the bit time); the Bit Timing Logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

21.3.2.1 Calculation of the Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting Bit time (1 / Bit rate) must be an integer multiple of the CAN clock period.

NOTE: 8 MHz is the minimum CAN clock frequency required to operate the DCAN at a bit rate of 1 MBit/s.

The bit time may consist of 8 to 25 time quanta. The length of the time quantum t_q is defined by the Baud Rate Prescaler with t_q = (Baud Rate Prescaler) / CAN_CLK. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

First part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded up to the nearest integer multiple of t_q).

The Sync_Seg is 1 t_q long (fixed), leaving (bit time – Prop_Seg – 1) t_q for the two Phase Buffer Segments. If the number of remaining t_q is even, the Phase Buffer Segments have the same length, Phase_Seg2 = Phase_Seg1, else Phase_Seg2 = Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than any CAN controller's Information Processing Time in the network, which is device dependent and can be in the range of [0 … 2] t_q.

The length of the Synchronization Jump Width is set to its maximum value, which is the minimum of 4 and Phase_Seg1.

If more than one configurations are possible to reach a certain Bit rate, it is recommended to choose the configuration that allows the highest oscillator tolerance range.

CAN nodes with different clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The CAN system's oscillator tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the oscillator frequencies' stability has to be increased in order to find a protocol compliant configuration of the CAN bit timing.

The resulting configuration is written into the Bit Timing Register:

Tseg2 = Phase_Seg2 - 1
Tseg1 = Phase_Seg1 + Prop_Seg - 1
SJW = SynchronizationJumpWidth - 1
BRP = Prescaler - 1

21.3.2.2 Calculation of BRP Values

If Baud and CAN_CLK(VCLK) are already known, the BRP/BRPE values need to be calculated to be programmed into the register. It is calculated using the following equation:

BRP = CAN_CLK / (BAUD)(1 + TSEG1 + TSEG2) (7)

21.3.2.3 Example for Bit Timing at High Baudrate

In this example, the frequency of CAN_CLK is 10 MHz, BRP is 0, the bit rate is 1 MBit/s.

t_q	100	ns	=	t_{CAN_CLK}
delay of bus driver	60	ns		
delay of receiver circuit	40	ns		
delay of bus line (40m)	220	ns		
t_{Prop}	700	ns	=	INT (2 × delays + 1) = 7 • t_q
t_{SJW}	100	ns	=	1 × t_q
t_{TSeg1}	800	ns	=	t_{Prop} + t_{SJW}
t_{TSeg2}	100	ns	=	Information Processing Time + 1 • t_q
$t_{Sync\text{-}Seg}$	100	ns	=	1 × t_q
bit time	1000	ns	=	$t_{Sync\text{-}Seg}$ + t_{TSeg1} + t_{TSeg2}
tolerance for CAN_CLK	1.58	%	=	$\frac{\min(TSeg1, TSeg2)}{[2 \times (13 \times bit\_time - TSeg2)]}$ (8)
			=	$\frac{0.1\mu s}{[2 \times (13 \times 1\mu s - 0.1\mu s)]}$ (9)
			=	0.38%

In this example, the concatenated bit time parameters are $(1\text{-}1)_3$ & $(8\text{-}1)_4$ & $(1\text{-}1)_2$ & $(1\text{-}1)_6$, so the Bit Timing Register is programmed to 0000 0700h.

21.3.2.4 Example for Bit Timing at Low Baudrate

In this example, the frequency of CAN_CLK is 2 MHz, BRP is 1, the bit rate is 100 KBit/s.

t_q	100	ns	=	$2 \times t_{CAN\_CLK}$
delay of bus driver	200	ns		
delay of receiver circuit	80	ns		
delay of bus line (40m)	220	ns		
t_{Prop}	1	µs	=	$1 \times t_q$
t_{SJW}	4	µs	=	$4 \times t_q$
t_{TSeg1}	5	µs	=	$t_{Prop} + t_{SJW}$
t_{TSeg2}	3	µs	=	Information Processing Time + $3 \times t_q$
$t_{Sync\text{-}Seg}$	1	µs	=	$1 \times t_q$
bit time	9	µs	=	$t_{Sync\text{-}Seg} + t_{TSeg1} + t_{TSeg2}$
tolerance for CAN_CLK	0.43	%	=	$\frac{\min(TSeg1, TSeg2)}{[2 \times (13 \times bit\_time - TSeg2)]}$ (10)
			=	$\frac{3\mu s}{[2 \times (13 \times 9\mu s - 3\mu s)]}$ (11)
			=	1.32%

In this example, the concatenated bit time parameters are $(3\text{-}1)_3$ & $(5\text{-}1)_4$ & $(4\text{-}1)_2$ & $(2\text{-}1)_6$, so the Bit Timing Register is programmed to 0000 24C1h.

21.4 CAN Module Configuration

After a hardware reset all CAN protocol functions are disabled.The CAN module must be initialized and configured before it can participate on the CAN bus.

21.4.1 DCAN RAM Initialization Through Hardware

To start with a clean DCAN RAM ,the complete DCAN RAM has to be initialized with zeros and the ECC bits set accordingly by configuring the following registers in the system module:

1. Memory Hardware Initialization Global Control Register (MINITGCR)
2. Memory Initialization Enable Register (MSINENA)

For more details on RAM hardware initialization support, refer to the system module reference guide.

21.4.2 CAN Module Initialization

To initialize the CAN Controller, you have to set up the CAN Bit timing and those message objects that have to be used for CAN communication. Message objects that are not needed, can be deactivated.

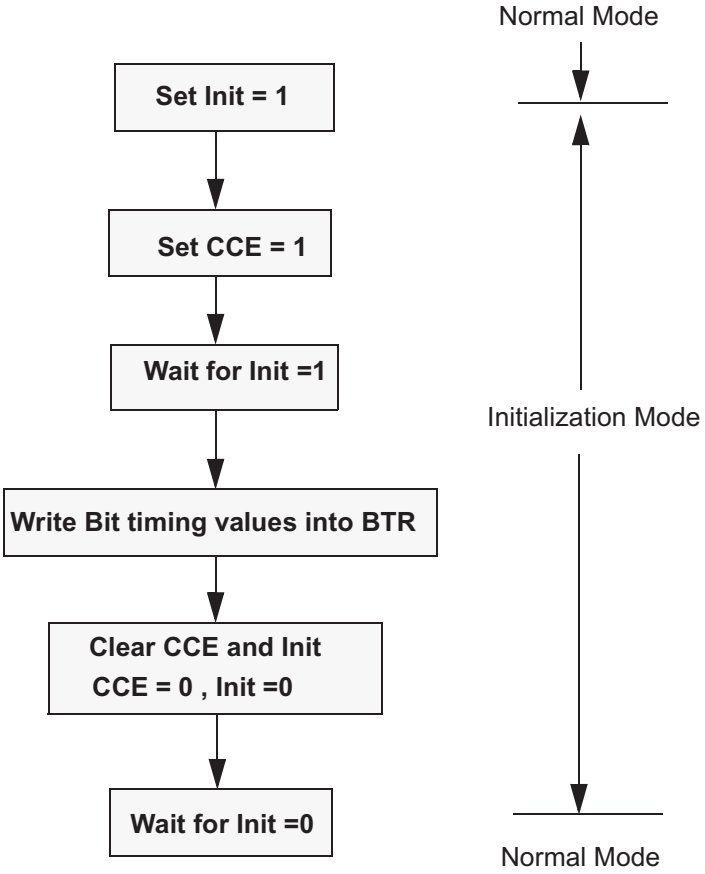
So the two critical steps are:

1. Configuration of CAN Bit Timings
2. Configuration of Message Objects

21.4.2.1 Software Configuration of CAN Bit Timings

This step involves configuring the CAN baud rate register with the calculated CAN bit timing value. The calculation procedure of CAN bit timing values for BTR register are mentioned in Section 21.3. Refer to Figure 21-3 for CAN bit timing software configuration flow.

Figure 21-3. CAN Bit-timing Configuration



Step 1: Enter “initialization mode” by setting the Init (Initialization) bit in the CAN Control Register.

While the Init bit is set, the message transfer from and to the CAN bus is stopped, and the status of the CAN_TX output is recessive (high).

The CAN error counters are not updated. Setting the Init bit does not change any other configuration register.

Also note that the CAN module is also in initialization mode on hardware reset and during Bus-Off.

Step 2: Set the CCE (Configure Change Enable) bit in the CAN Control Register.

The access to the Bit Timing Register for the configuration of the Bit timing is enabled when both Init and CCE bits in the CAN Control Register are set.

Step 3: Wait for the Init bit to get set. This would make sure that the module has entered “Initialization mode”.

Step 4: Write the Bit-Timing values into the Bit-Timing Register (BTR).

Refer to Section 21.3.2.1 for BTR value calculation for a given bit-timing.

Step 5: Clear the CCE bit followed by Init bit.

Step 6: Wait for the Init bit to clear. This would make sure that the module has come out of “initialization mode”.

After step 6 (Init bit cleared), the module will attempt a synchronization on the CAN bus, provided that the BTR settings are meeting the CAN bus parameters.

NOTE: The module would not come out of the “initialization mode” if any incorrect BTR values are written in step 4.

21.4.2.2 Configuration of Message Objects

The whole Message RAM should be configured before putting the CAN into operation. All the message objects are deactivated by default. You should configure the message object that are to be used to a particular identifier. you can change the configuration of any message object or deactivate it when required.

The message objects can be configured only through the Interface registers (IFx) and the CPU does not have direct access to the message object (Message RAM) when DCAN is in operation.

To configure the message objects, you must know about:

1. The message object structure (Section 21.5)
2. The interface register set (IFx) (Section 21.6)

NOTE: The message objects initialization is independent of the bit-timing configuration procedure.

21.5 Message RAM

The DCAN Message RAM contains message objects and ECC bits for the message objects.

21.5.1 Structure of Message Objects

Figure 21-4 shows the structure of a message object.

The grayed fields are those parts of the message object that are represented in dedicated registers. For example, the transmit request flags of all message objects are represented in centralized transmit request registers.

Figure 21-4. Structure of a Message Object

Message Object												
UMask	Msk[28:0]	MXtd	MDir	EoB	unused	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID[28:0]	Xtd	Dir	DLC[3:0]	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7

Table 21-2. Message Object Field Descriptions

Name	Value	Description
MsgVal		Message valid
	0	The message object is ignored by the Message Handler.
	1	The message object is to be used by the Message Handler.
		Note: The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN Control Register. MsgVal must also be reset if the messages object is no longer used in operation. For reconfiguration of message objects during normal operation see Section 21.7.6 and Section 21.7.7.
UMask		Use Acceptance Mask
	0	Mask bits (Msk[28:0], MXtd and MDir) are ignored and not used for acceptance filtering.
	1	Mask bits are used for acceptance filtering.
		Note: If the UMask bit is set to 1, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to 1.
ID[28:0]		Message Identifier
	ID[28:0]	29-bit ("extended") identifier bits
	ID[28:18]	11-bit ("standard") identifier bits
Msk[28:0]		Identifier Mask
	0	The corresponding bit in the message identifier is not used for acceptance filtering (don't care).
	1	The corresponding bit in the message identifier is used for acceptance filtering.
Xtd		Extended Identifier
	0	The 11-bit ("standard") identifier will be used for this message object.
	1	The 29-bit ("extended") identifier will be used for this message object.
MXtd		Mask Extended Identifier
	0	The extended identifier bit (IDE) has no effect on the acceptance filtering.
	1	The extended identifier bit (IDE) is used for acceptance filtering.
		Note: When 11-bit ("standard") identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.
Dir		Message Direction
	0	Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, the message is stored in this message object.
	1	Direction = transmit: On TxRqst, a data frame is transmitted. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
MDir		Mask Message Direction
	0	The message direction bit (Dir) has no effect on the acceptance filtering.
	1	The message direction bit (Dir) is used for acceptance filtering.

Table 21-2. Message Object Field Descriptions (continued)

Name	Value	Description
EOB		End of Block
	0	The message object is part of a FIFO Buffer block and is not the last message object of this FIFO Buffer block.
	1	The message object is a single message object or the last message object in a FIFO Buffer block.
		Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to 1.
NewDat		New Data
	0	No new data has been written into the data bytes of this message object by the Message Handler since the last time when this flag was cleared by the CPU.
	1	The Message Handler or the CPU has written new data into the data bytes of this message object.
MsgLst		Message Lost (only valid for message objects with direction = receive)
	0	No message was lost since the last time when this bit was reset by the CPU.
	1	The Message Handler stored a new message into this message object when NewDat was still set, so the previous message has been overwritten.
RxIE		Receive Interrupt Enable
	0	IntPnd will not be triggered after the successful reception of a frame.
	1	IntPnd will be triggered after the successful reception of a frame.
TxIE		Transmit Interrupt Enable
	0	IntPnd will not be triggered after the successful transmission of a frame.
	1	IntPnd will be triggered after the successful transmission of a frame.
IntPnd		Interrupt Pending
	0	This message object is not the source of an interrupt.
	1	This message object is the source of an interrupt. The Interrupt Identifier in the Interrupt Register will point to this message object if there is no other interrupt source with higher priority.
RmtEn		Remote Enable
	0	At the reception of a Remote Frame, TxRqst is not changed.
	1	At the reception of a Remote Frame, TxRqst is set.
TxRqst		Transmit Request
	0	This message object is not waiting for a transmission.
	1	The transmission of this message object is requested and is not yet done.
DLC[3:0]		Data Length Code
	0-8	Data Frame has 0-8 data bytes.
	9-15	Data Frame has 8 data bytes.
		Note: The Data Length Code of a message object must be defined to the same value as in the corresponding objects with the same identifier at other nodes. When the Message Handler stores a data frame, it will write the DLC to the value given by the received message.
Data 0		1st data byte of a CAN Data Frame
Data 1		2nd data byte of a CAN Data Frame
Data 2		3rd data byte of a CAN Data Frame
Data 3		4th data byte of a CAN Data Frame
Data 4		5th data byte of a CAN Data Frame
Data 5		6th data byte of a CAN Data Frame
Data 6		7th data byte of a CAN Data Frame
Data 7		8th data byte of a CAN Data Frame
		Note: Byte Data 0 is the first data byte shifted into the shift register of the CAN Core during a reception, byte Data 7 is the last. When the Message Handler stores a data frame, it will write all the eight data bytes into a message object. If the Data Length Code is less than 8, the remaining bytes of the message object may be overwritten by undefined values.

21.5.2 Addressing Message Objects in RAM

The starting location of a particular message object in RAM is:

Message RAM base address + (message object number) × 0x20.

This means that Message Object 1 starts at offset 0x0020; Message Object 2 starts at offset 0x0040, and so on.

NOTE: 0 is not a valid message object number. At address 0x0000, message object number 64 is located. Writing to the address of an unimplemented message object may overwrite an implemented message object.

The base address for DCAN1 RAM is FF1E 0000h, DCAN2 RAM is FF1C 0000h, DCAN3 RAM is FF1A 0000h, and DCAN4 RAM base address is FF18 0000h.

Message Object number 1 has the highest priority.

Table 21-3. Message RAM Addressing in Debug/Suspend and RDA Mode

Message Object Number	Base Address Offset	Word Number	Debug/Suspend mode, see Section 21.5.3	RDA mode, see Section 21.5.4
1	0x0020	1	Reserved	Data Bytes 4-7
	0x0024	2	MXtd, MDir, Mask	Data Bytes 0-3
	0x0028	3	Xtd, Dir, ID	ID[27:0], DLC
	0x002C	4	Ctrl	Mask, Xtd, Dir, ID[28]
	0x0030	5	Data Bytes 3-0	Reserved, Ctrl, MXtd, MDir
	0x0034	6	Data Bytes 7-4	--
:	:	:	:	:
31	0x03E0	1	Reserved	Data Bytes 4-7
	0x03E4	2	MXtd, MDir, Mask	Data Bytes 0-3
	0x03E8	3	Xtd, Dir, ID	ID[27]:0, DLC
	0x03EC	4	Ctrl	Mask, Xtd, Dir, ID[28]
	0x03F0	5	Data Bytes 3-0	Reserved, Ctrl, MXtd, MDir
	0x03F4	6	Data Bytes 7-4	--
:	:	:	:	:
63	0x07E0	1	Reserved	Data Bytes 4-7
	0x07E4	2	MXtd, MDir, Mask	Data Bytes 0-3
	0x07E8	3	Xtd, Dir, ID	ID[27:0], DLC
	0x07EC	4	Ctrl	Mask, Xtd, Dir, ID[28]
	0x07F0	5	Data Bytes 3-0	Reserved, Ctrl, MXtd, MDir
	0x07F4	6	Data Bytes 7-4	--
64	0x0000	1	Reserved	Data Bytes 4-7
	0x0004	2	MXtd, MDir, Mask	Data Bytes 0-3
	0x0008	3	Xtd, Dir, ID	ID[27]:0, DLC
	0x000C	4	Ctrl	Mask, Xtd, Dir, ID[28]
	0x0010	5	Data Bytes 3-0	Reserved, Ctrl, MXtd, MDir
	0x0014	6	Data Bytes 7-4	--

21.5.3 Message RAM Representation in Debug/Suspend Mode

In Debug/Suspend mode, the Message RAM will be memory mapped. This allows the external debug unit to access the Message RAM.

NOTE: During Debug/Suspend Mode, the Message RAM cannot be accessed via the IFx register sets.

Figure 21-5. Message RAM Representation in Debug/Suspend Mode

Bit	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
MsgAddr + 0x00	Reserved															
	Reserved											Reserved				
MsgAddr + 0x04	MXtd	MDir	Rsvd	Msk[28:16]												
	Msk[15:0]															
MsgAddr + 0x08	Rsvd	Xtd	Dir	ID[28:16]												
	ID[15:0]															
MsgAddr + 0x0C	Reserved															
	Rsvd	MsgLst	Rsvd	UMask	TxIE	RxIE	RmtEn	Rsvd	EOB	Reserved			DLC[3:0]			
MsgAddr + 0x10	Data 3								Data 2							
	Data 1								Data 0							
MsgAddr + 0x14	Data 7								Data 6							
	Data 5								Data 4							

21.5.4 Message RAM Representation in Direct Access Mode

When the RDA bit in Test Register is set while the DCAN module is in Test Mode (Test bit in CAN control register is set), the CPU has direct access to the Message RAM. Due to the 32-bit bus structure, the RAM is split into word lines to support this feature. The CPU has access to one word line at a time only.

In RAM Direct Access mode, the RAM is represented by a continuous memory space within the address frame of the DCAN module, starting at the Message RAM base address.

NOTE: During Direct Access Mode, the Message RAM cannot be accessed via the IFx register sets. Before entering RDA mode, it must be ensured that the Init bit is set to avoid any conflicts with the message handler accessing the message RAM.

Any read or write to the RAM addresses for RamDirectAccess during normal operation mode (TestMode bit or RDA bit is not set) will be ignored.

Writes to Reserved bits have no effect.

Figure 21-6. Message RAM Representation in RAM Direct Access Mode

Bit	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
MsgAddr + 0x00	Data 4								Data 5							
	Data 6								Data 7							
MsgAddr + 0x04	Data 0								Data 1							
	Data 2								Data 3							
MsgAddr + 0x08	ID[27:12]															
	ID[11:0]												DLC[3:0]			
MsgAddr + 0x0C	Msk[28:13]															
	Msk[12:0]													Xtd	Dir	ID[28]
MsgAddr + 0x10	Reserved															
	Reserved				Reserved				MsgLst	UMask	TxIE	RxIE	RmtEn	EOB	MXtd	MDir

21.5.5 ECC RAM

On devices with SECDED implementation for the message RAM, the ECC bits are stored in a dedicated ECC RAM area that is memory-mapped as follows: The location of the ECC bits for a particular message object in RAM is: Message RAM base address + 0x1000 + (message object number) * 0x20.

NOTE: '0' is not a valid message object number. At address 0x1000, the ECC bits of the last implemented message object are located.

As shown in Figure 21-7, the ECC bits for the last implemented Message Object (here: 128) are located at offset 0x1000; the ECC bits for Message Object 1 are located at offset 0x1020, and the ECC bits for Message Object 127 are located at offset 0x1FE0. The ECC RAM is only memory mapped if SECDED diagnostic mode is enabled.

Figure 21-7. ECC RAM Representation

Bit	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
Msg RAM base + 0x1000	Reserved															
	Reserved							ECC[8:0] last implemented Message Object (here: 128)								
Msg RAM base + 0x1020	Reserved															
	Reserved							ECC[8:0] Message Object 1								
:																
Msg RAM base + 0x1FE0	Reserved															
	Reserved							ECC[8:0] Message Object 127								

21.6 Message Interface Register Sets

Accesses to the Message RAM are performed via the Interface Register sets:

- Interface Register 1 and 2 (IF1 and IF2)
- Interface Register 3 (IF3)

The IF3 register set can be configured to automatically receive control and user data from the Message RAM when a message object has been updated after reception of a CAN message. The CPU does not need to initiate the transfer from Message RAM to IF3 register set.

The Message Handler avoids potential conflicts between concurrent accesses to Message RAM and CAN frame reception/transmission.

There are two modes where the Message RAM can be directly accessed by the CPU:

1. In Debug/Suspend mode (see Section 21.5.3)
2. In RAM Direct Access (RDA) mode (see Section 21.5.4)

For the Message RAM Base address, refer to the device datasheet.

A complete message object (see Section 21.5.1) or parts of the message object may be transferred between the Message RAM and the IF1/IF2 Register set (see) in one single transfer.

21.6.1 Message Interface Register Sets 1 and 2

The Interface Register sets IF1 and IF2 provide indirect read/write access from the CPU to the Message RAM. The IF1 and IF2 register sets can buffer control and user data to be transferred to and from the message objects.

Table 21-4. Message Interface Register Sets 1 and 2

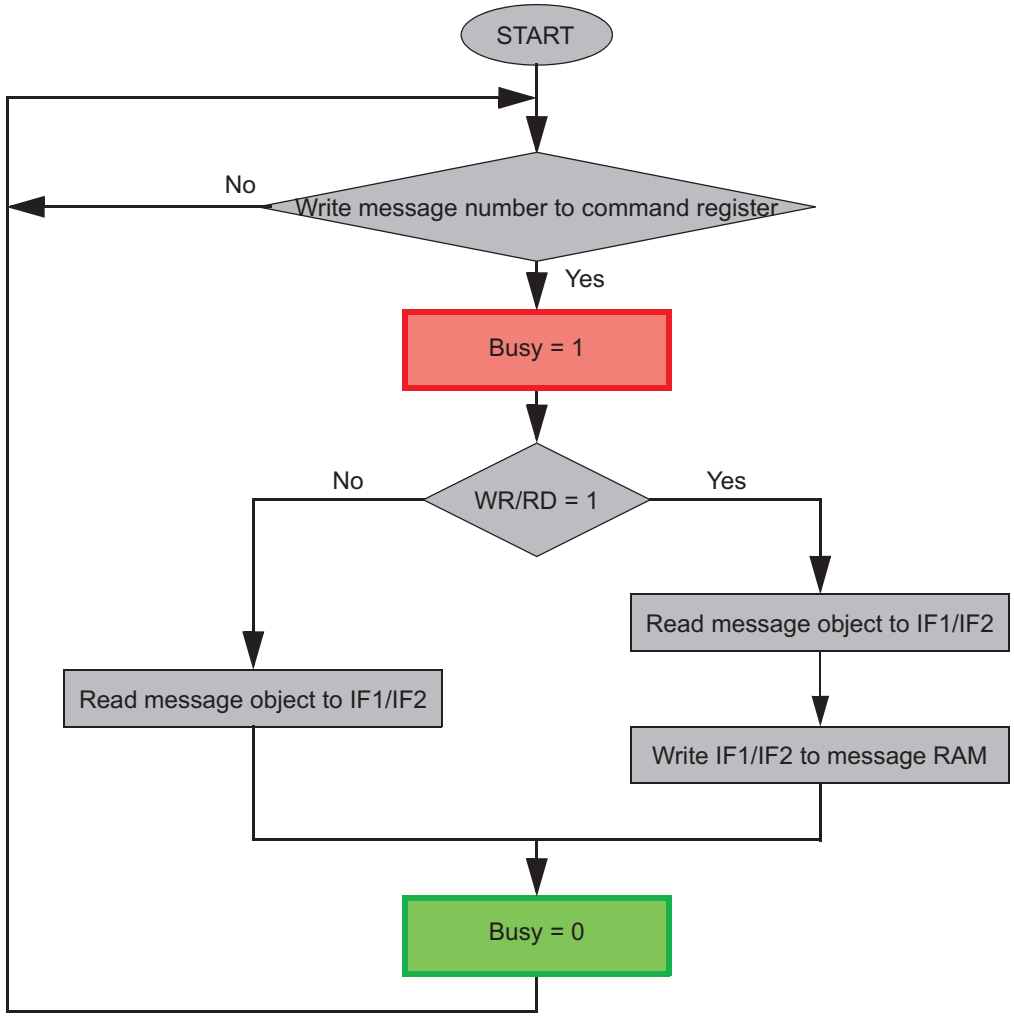
Address	IF1 Register Set		Address	IF2 Register Set	
[CAN Base +]	31 16	15 0	[CAN Base +]	31 16	15 0
0x100	IF1 Command Mask	IF1 Command Request	0x120	IF2 Command Mask	IF2 Command Request
0x104	IF1 Mask 2	IF1 Mask 1	0x124	IF2 Mask 2	IF2 Mask 1
0x108	IF1 Arbitration 2	IF1 Arbitration 1	0x128	IF2 Arbitration 2	IF2 Arbitration 1
0x10C	Rsvd	IF1 Message Control	0x12C	Rsvd	IF2 Message Control
0x110	IF1 Data A 2	IF1 Data A 1	0x130	IF2 Data A 2	IF2 Data A 1
0x114	IF1 Data B 2	IF1 Data B 1	0x134	IF2 Data B 2	IF2 Data B 1

21.6.2 Using Message Interface Register Sets 1 and 2

The Command Register addresses the desired message object in the Message RAM and specifies whether a complete message object or only parts should be transferred. The data transfer is initiated by writing the message number to the bits [7:0] of the Command Register.

When the CPU initiates a data transfer between the IF1/IF2 Registers and Message RAM, the Message Handler sets the Busy bit in the respective Command Register to '1'. After the transfer has completed, the Busy bit is set back to '0' (see Figure 21-8).

Figure 21-8. Data Transfer Between IF1 / IF2 Registers and Message RAM



21.6.3 Message Interface Register 3

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from Message RAM by CPU. The intention of this feature of IF3 is to provide an interface for the DMA to read packets efficiently.

Table 21-5. Message Interface Register 3

Address	IF3 Register Set	
[CAN Base +]	31 16	15 0
0x140	reserved	IF3 Observation
0x144	IF3 Mask 2	IF3 Mask 1
0x148	IF3 Arbitration 2	IF3 Arbitration 1
0x14C	reserved	IF3 Message Control
0x150	IF3 Data A 2	IF3 Data A 1
0x154	IF3 Data B 2	IF3 Data B 1
:	:	:
0x160	IF3 Update Enable 2	IF3 Update Enable 1
0x164	IF3 Update Enable 4	IF3 Update Enable 3
0x168	IF3 Update Enable 6	IF3 Update Enable 5
0x16C	IF3 Update Enable 8	IF3 Update Enable 7

The automatic update functionality can be programmed for each message object (see IF3 Update Enable Register,).

All valid message objects in Message RAM that are configured for automatic update, will be checked for active NewDat flags. If such a message object is found, it will be transferred to the IF3 register (if no previous DMA transfers are ongoing), controlled by IF3 Observation register. If more than one NewDat flag is active, the message object with the lowest number has the highest priority for automatic IF3 update.

The NewDat bit in the message object will be reset by a transfer to IF3.

If DCAN internal IF3 update is complete, a DMA request is generated. The DMA request stays active until first read access to one of the IF3 registers. The DMA functionality has to be enabled by setting bit DE3 in CAN Control register. Please refer to the device datasheet to find out if this DMA source is available.

NOTE: The IF3 register set can not be used for transferring data into message objects.

21.7 Message Object Configurations

This section describes the possible message object configurations for CAN communication.

21.7.1 Configuration of a Transmit Object for Data Frames

Figure 21-9 shows how a Transmit Object can be initialized.

Figure 21-9. Initialization of a Transmit Object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The Arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of the outgoing message. If an 11-bit Identifier (Standard Frame) is used (Xtd = 0), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored.

The Data Registers (DLC[3:0] and Data0-7) are given by the application, TxRqst and RmtEn should not be set before the data is valid.

If the TxIE bit is set, the IntPnd bit will be set after a successful transmission of the message object.

If the RmtEn bit is set, a matching received Remote Frame will cause the TxRqst bit to be set; the Remote Frame will autonomously be answered by a Data Frame.

The Mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask = 1) to allow groups of Remote Frames with similar identifiers to set the TxRqst bit. The Dir bit should not be masked. For details, see Section 21.8.8. Identifier masking must be disabled (UMask = 0) if no Remote Frames are allowed to set the TxRqst bit (RmtEn = 0).

21.7.2 Configuration of a Transmit Object for Remote Frames

It is not necessary to configure Transmit Objects for the transmission of Remote Frames. Setting TxRqst for a Receive Object will cause the transmission of a Remote Frame with the same identifier as the Data Frame for which this receive Object is configured.

21.7.3 Configuration of a Single Receive Object for Data Frames

Figure 21-10 shows how a Receive Object for Data Frames can be initialized.

Figure 21-10. Initialization of a Single Receive Object for Data Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The Arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (Standard Frame) is used (Xtd = 0), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a Data Frame with an 11-bit Identifier is received, ID[17:0] will be set to 0.

The Data Length Code (DLC[3:0]) is given by the application. When the Message Handler stores a Data Frame in the message object, it will store the received Data Length Code and eight data bytes. If the Data Length Code is less than 8, the remaining bytes of the message object may be overwritten by non specified values.

The Mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask = '1') to allow groups of Data Frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. If some bits of the Mask bits are set to "don't care", the corresponding bits of the Arbitration Register will be overwritten by the bits of the stored Data Frame.

If the RxIE bit is set, the IntPnd bit will be set when a received Data Frame is accepted and stored in the message object.

If the TxRqst bit is set, the transmission of a Remote Frame with the same identifier as actually stored in the Arbitration bits will be triggered. The content of the Arbitration bits may change if the Mask bits are used (UMask = 1 for acceptance filtering.

21.7.4 Configuration of a Single Receive Object for Remote Frames

Figure 21-11 shows how a Receive Object for Remote Frames can be initialized.

Figure 21-11. Initialization of a Single Receive Object for Remote Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	appl.	0	0	0	0

Receive Objects for Remote Frames may be used to monitor Remote Frames on the CAN bus. The Remote Frame stored in the Receive Object will not trigger the transmission of a Data Frame. Receive Objects for Remote Frames may be expanded to a FIFO buffer, see Section 21.7.5.

UMask must be set to 1. The Mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be set to “must-match” or to “don’t care”, to allow groups of Remote Frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. For details, see Section 21.8.8.

The Arbitration bits (ID[28:0] and Xtd bit) may be given by the application. They define the identifier and type of accepted received Remote Frames. If some bits of the Mask bits are set to “don’t care”, the corresponding bits of the Arbitration bits will be overwritten by the bits of the stored Remote Frame. If an 11-bit Identifier (Standard Frame) is used (Xtd = 0), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a Remote Frame with an 11-bit Identifier is received, ID[17:0] will be set to 0.

The Data Length Code (DLC[3:0]) may be given by the application. When the Message Handler stores a Remote Frame in the message object, it will store the received Data Length Code. The data bytes of the message object will remain unchanged.

If the RxIE bit is set, the IntPnd bit will be set when a received Remote Frame is accepted and stored in the message object.

21.7.5 Configuration of a FIFO Buffer

With the exception of the EoB bit, the configuration of Receive Objects belonging to a FIFO Buffer is the same as the configuration of a single Receive Object.

To concatenate multiple message objects to a FIFO Buffer, the identifiers and masks (if used) of these message objects have to be programmed to matching values. Due to the implicit priority of the message objects, the message object with the lowest number will be the first message object of the FIFO Buffer. The EoB bit of all message objects of a FIFO Buffer except the last one have to be programmed to zero. The EoB bits of the last message object of a FIFO Buffer is set to one, configuring it as the end of the block.

21.7.6 Reconfiguration of Message Objects for the Reception of Frames

A message object with Dir = ‘0’ is configured for the reception of data frames, with Dir = ‘1’ AND Umask = ‘1’ AND RmtEn = ‘0’ it is configured for the reception of remote frames.

It is neccessary to reset MsgVal to not valid before changing any of the following configuration and control bits: ID[28:0], Xtd, Dir, DLC[3:0], RxIE, TxIE, RmtEn, EoB, Umask, Msk[28:0], MXtd, and MDir.

These parts of a message object may be changed without clearing MsgVal: Data[7:0], TxRqst, NewDat, MsgLst, and IntPnd.

21.7.7 Reconfiguration of Message Objects for the Reception of Frames

A message object with Dir = ‘1’ AND (Umask = ‘0’ OR RmtEn = ‘1’) is configured for the transmission of data frames.

It is neccessary to reset MsgVal to not valid before changing any of the following configuration and control bits: Dir, RxIE, TxIE, RmtEn, EoB, Umask, Msk[28:0], MXtd, and MDir

These parts of a message object may be changed without clearing MsgVal: ID[28-0], Xtd, DLC[3:0], Data[7:0], TxRqst, NewDat, MsgLst, and IntPnd.

21.8 Message Handling

When initialization is finished, the DCAN module synchronizes itself to the traffic on the CAN bus. It does acceptance filtering on received messages and stores those frames that are accepted into the designated message objects.

The application has to update the data of the messages to be transmitted and enable and request their transmission. The transmission is requested automatically when a matching Remote Frame is received.

The application may read messages that are received and accepted. Messages that are not read before the next messages is accepted for the same message object will be overwritten.

Messages may be read based on interrupts or by polling.

21.8.1 Message Handler Overview

The Message Handler state machine controls the data transfer between the Rx/Tx Shift Register of the CAN Core and the Message RAM. It performs the following tasks:

- Data Transfer from Message RAM to CAN Core (messages to be transmitted).
- Data Transfer from CAN Core to the Message RAM (received messages).
- Data Transfer from CAN Core to the Acceptance Filtering unit.
- Scanning of Message RAM for a matching message object (acceptance filtering).
- Scanning the same message object after being changed by IF1/IF2 registers when priority is same or higher as message the object found by last scanning.
- Handling of TxRqst flags.
- Handling of interrupt flags.

The Message Handler registers contains status flags of all message objects grouped into the following topics:

- Transmission Request flags
- New Data flags
- Interrupt Pending Flags
- Message Valid Registers

Instead of collecting the listed status information of each message object via IFx registers separately, these Message Handler registers provides a fast and easy way to get an overview (for example, about all pending transmission requests).

All Message Handler registers are read-only.

21.8.2 Receive/Transmit Priority

The receive/transmit priority for the message objects is attached to the message number, not to the CAN identifier. Message object 1 has the highest priority, while the last implemented message object has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding message object, so messages with the highest priority can be placed in the message objects with the lowest numbers.

The acceptance filtering for received Data Frames or Remote Frames is also done in ascending order of message objects, so a frame that has been accepted by a message object cannot be accepted by another message object with a higher Message Number. The last message object may be configured to accept any Data Frame or Remote Frame that was not accepted by any other message object, for nodes that need to log the complete message traffic on the CAN bus.

21.8.3 Transmission of Messages in Event Driven CAN Communication

If the shift register of the CAN Core is ready for loading and if there is no data transfer between the IFx Registers and Message RAM, the d bits in the Message Valid Register and the TxRqst bits in the Transmission Request Register are evaluated. The valid message object with the highest priority pending transmission request is loaded into the shift register by the Message Handler and the transmission is started. The message object's NewDat bit is reset.

After a successful transmission and if no new data was written to the message object (NewDat = '0') since the start of the transmission, the TxRqst bit will be reset. If TxIE is set, IntPnd will be set after a successful transmission. If the DCAN has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. If meanwhile the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

If Automatic Retransmission mode is disabled by setting the DAR bit in the CAN Control Register, the behavior of bits TxRqst and NewDat in the Message Control Register of the Interface Register set is as follows:

- When a transmission starts, the TxRqst bit of the respective Interface Register set is reset, while bit NewDat remains set.
- When the transmission has been successfully completed, the NewDat bit is reset.

When a transmission failed (lost arbitration or error) bit NewDat remains set. To restart the transmission, the application has to set TxRqst again.

Received Remote Frames do not require a Receive Object. They will automatically trigger the transmission of a Data Frame, if in the matching Transmit Object the RmtEn bit is set.

21.8.4 Updating a Transmit Object

The CPU may update the data bytes of a Transmit Object any time via the IF1/IF2 Interface Registers, neither d nor TxRqst have to be reset before the update.

Even if only a part of the data bytes are to be updated, all four bytes in the corresponding IF1/IF2 Data A Register or IF1/IF2 Data B Register have to be valid before the content of that register is transferred to the message object. Either the CPU has to write all four bytes into the IF1/IF2 Data Register or the message object is transferred to the IF1/IF2 Data Register before the CPU writes the new data bytes.

When only the data bytes are updated, first 0x87 can be written to bits [23:16] of the Command Register and then the number of the message object is written to bits [7:0] of the Command Register, concurrently updating the data bytes and setting TxRqst with NewDat.

To prevent the reset of TxRqst at the end of a transmission that may already be in progress while the data is updated, NewDat has to be set together with TxRqst in event driven CAN communication. For details, see Section 21.8.3.

When NewDat is set together with TxRqst, NewDat will be reset as soon as the new transmission has started.

21.8.5 Changing a Transmit Object

If the number of implemented message objects is not sufficient to be used as permanent message objects only, the Transmit Objects may be managed dynamically. The CPU can write the whole message (Arbitration, Control, and Data) into the Interface Register. The bits [23:16] of the Command Register can be set to 0xB7 for the transfer of the whole message object content into the message object. Before changing the configuration of a message object, MsgVal has to be reset (see Section 21.7.7).

If a previously requested transmission of this message object is not completed but already in progress, it will be continued; however it will not be repeated if it is disturbed.

To only update the data bytes of a message to be transmitted, bits [23:16] of the Command Register should be set to 0x87.

NOTE: After the update of the Transmit Object, the Interface Register set will contain a copy of the actual contents of the object, including the part that had not been updated.

21.8.6 Acceptance Filtering of Received Messages

When the arbitration and control bits (Identifier + IDE + RTR + DLC) of an incoming message is completely shifted into the shift register of the CAN Core, the Message Handler starts to scan of the Message RAM for a matching valid message object:

- The Acceptance Filtering unit is loaded with the arbitration bits from the CAN Core shift register.
- Then the arbitration and mask bits (including MsgVal, UMask, NewDat, and EoB) of Message Object 1 are loaded into the Acceptance Filtering unit and are compared with the arbitration bits from the shift register. This is repeated for all following message objects until a matching message object is found, or until the end of the Message RAM is reached.
- If a match occurs, the scanning is stopped and the Message Handler proceeds depending on the type of the frame (Data Frame or Remote Frame) received.

21.8.7 Reception of Data Frames

The Message Handler stores the message from the CAN Core shift register into the respective message object in the Message RAM. Not only the data bytes, but all arbitration bits and the Data Length Code are stored into the corresponding message object. This ensures that the data bytes stay associated to the identifier even if arbitration mask registers are used.

The NewDat bit is set to indicate that new data (not yet seen by the CPU) has been received. The CPU should reset the NewDat bit when it reads the message object. If at the time of the reception the NewDat bit was already set, MsgLst is set to indicate that the previous data (supposedly not seen by the CPU) is lost. If the RxIE bit is set, the IntPnd bit is set, causing the Interrupt Register to point to this message object.

The TxRqst bit of this message object is reset to prevent the transmission of a Remote Frame, while the requested Data Frame has just been received.

21.8.8 Reception of Remote Frames

When a Remote Frame is received, three different configurations of the matching message object have to be considered:

1. Dir = 1 (direction = transmit), RmtEn = 1, UMask = 1 or 0: The TxRqst bit of this message object is set at the reception of a matching Remote Frame. The rest of the message object remains unchanged.
2. Dir = 1 (direction = transmit), RmtEn = 0, UMask = 0: The Remote Frame is ignored, this message object remains unchanged.
3. Dir = 1 (direction = transmit), RmtEn = 0, UMask = 1: The Remote Frame is treated similar to a received Data Frame. At the reception of a matching Remote Frame, the TxRqst bit of this message object is reset. The arbitration and control bits (Identifier + IDE + RTR + DLC) from the shift register are stored in the message object in the Message RAM and the NewDat bit of this message object is set. The data bytes of the message object remain unchanged.

21.8.9 Reading Received Messages

The CPU may read a received message any time via the IFx Interface Registers, the data consistency is guaranteed by the Message Handler state machine.

Typically the CPU will write first 0x7F to bits [23:16] and then the number of the message object to bits [7:0] of the Command Register. That combination will transfer the whole received message from the Message RAM into the Interface Register set. Additionally, the bits NewDat and IntPnd are cleared in the Message RAM (not in the Interface Register set). The values of these bits in the Message Control Register always reflect the status before resetting the bits.

If the message object uses masks for acceptance filtering, the arbitration bits show which of the different matching messages has been received.

The actual value of NewDat shows whether a new message has been received since last time when this message object was read. The actual value of MsgLst shows whether more than one message have been received since the last time when this message object was read. MsgLst will not be automatically reset.

21.8.10 Requesting New Data for a Receive Object

By means of a Remote Frame, the CPU may request another CAN node to provide new data for a receive object. Setting the TxRqst bit of a receive object will cause the transmission of a Remote Frame with the receive object's identifier. This Remote Frame triggers the other CAN node to start the transmission of the matching Data Frame. If the matching Data Frame is received before the Remote Frame could be transmitted, the TxRqst bit is automatically reset.

Setting the TxRqst bit without changing the contents of a message object requires the value 0x84 in bits [23:16] of the Command Register.

21.8.11 Storing Received Messages in FIFO Buffers

Several message objects may be grouped to form one or more FIFO Buffers. Each FIFO Buffer configured to store received messages with a particular (group of) Identifier(s). Arbitration and Mask Registers of the FIFO Buffer's message objects are identical. The EoB (End of Buffer) bits of all but the last of the FIFO Buffer's message objects are '0', in the last one the EoB bit is 1.

Received messages with identifiers matching to a FIFO Buffer are stored into a message object of this FIFO Buffer, starting with the message object with the lowest message number.

When a message is stored into a message object of a FIFO Buffer the NewDat bit of this message object is set. By setting NewDat while EoB is 0 the message object is locked for further write accesses by the Message Handler until the CPU has cleared the NewDat bit.

Messages are stored into a FIFO Buffer until the last message object of this FIFO Buffer is reached. If none of the preceding message objects is released by writing NewDat to 0, all further messages for this FIFO Buffer will be written into the last message object of the FIFO Buffer (EoB = 1) and therefore overwrite previous messages in this message object.

21.8.12 Reading from a FIFO Buffer

Several messages may be accumulated in a set of message objects that are concatenated to form a FIFO Buffer before the application program is required (in order to avoid the loss of data) to empty the buffer.

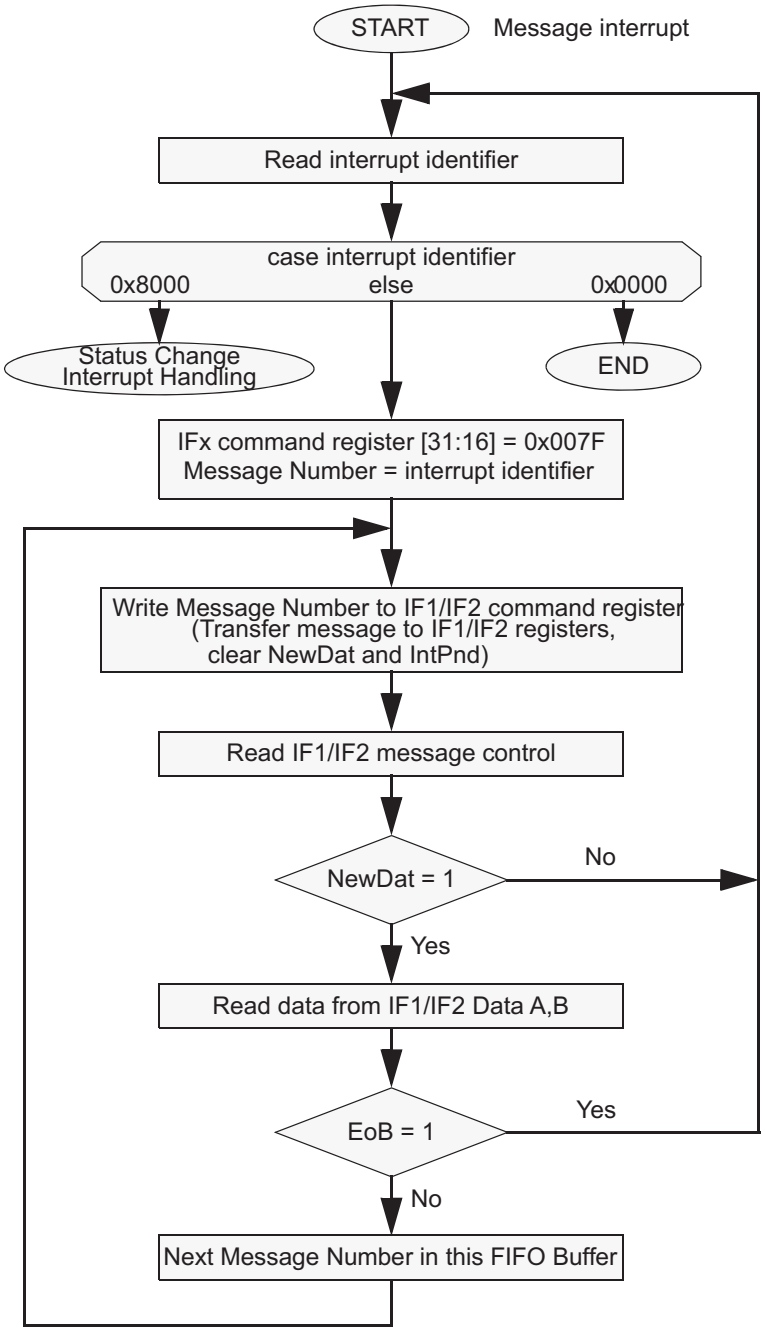
A FIFO Buffer of length N will store N-1 plus the last received message since last time it was cleared.

A FIFO Buffer is cleared by reading and resetting the NewDat bits of all its message objects, starting at the FIFO Object with the lowest message number. This should be done in a subroutine following the example shown in Figure 21-12.

NOTE: All message objects of a FIFO buffer needs to be read and cleared before the next batch of messages can be stored. Otherwise true FIFO functionality can not be guaranteed, since the message objects of a partly read buffer will be re-filled according to the normal (descending) priority.

Reading from a FIFO Buffer message object and resetting its NewDat bit is handled the same way as reading from a single message object.

Figure 21-12. CPU Handling of a FIFO Buffer (Interrupt Driven)



21.9 CAN Message Transfer

Once the DCAN is initialized and Init bit is reset to zero, the CAN Core synchronizes itself to the CAN bus and is ready for message transfer as per the configured message objects.

The CPU may enable the interrupt lines (setting IE0 and IE1 to 1) at the same time when it clears Init and CCE. The status interrupts EIE and SIE may be enabled simultaneously.

The CAN communication can be carried out in any of the following two modes:

1. Interrupt mode
2. Polling mode.

The Interrupt Register points to those message objects with IntPnd = 1. It is updated even if the interrupt lines to the CPU are disabled (IE0 / IE1 are zero).

The CPU may poll all Message Object's NewDat and TxRqst bits in parallel from the NewData X Registers and the Transmission Request X Registers. Polling can be made easier if all Transmit Objects are grouped at the low numbers, all Receive Objects are grouped at the high numbers.

Received messages are stored into their appropriate message objects if they pass acceptance filtering.

The whole message (including all arbitration bits, DLC and up to eight data bytes) is stored into the message object. As a consequence, when the identifier mask is used, the arbitration bits that are masked to "don't care" may change in the message object when a received message is stored.

The CPU may read or write each message at any time via the Interface Registers, as the Message Handler guarantees data consistency in case of concurrent accesses (for reconfiguration, see Section 21.7.6)

If a permanent message object (arbitration and control bits set up during configuration and leaving unchanged for multiple CAN transfers) exists for the message, it is possible to only update the data bytes.

If several transmit messages should be assigned to one message object, the whole message object has to be configured before the transmission of this message is requested.

The transmission of multiple message objects may be requested at the same time. They are subsequently transmitted, according to their internal priority.

Messages may be updated or set to not valid at any time, even if a requested transmission is still pending (for reconfiguration, see Section 21.7.7). However, the data bytes will be discarded if a message is updated before a pending transmission has started.

Depending on the configuration of the message object, a transmission may be automatically requested by the reception of a remote frame with a matching identifier.

21.9.1 Automatic Retransmission

According to the CAN Specification (ISO11898), the DCAN provides a mechanism to automatically retransmit frames that have lost arbitration or have been disturbed by errors during transmission. The frame transmission service will not be confirmed to you before the transmission is successfully completed.

By default, this automatic retransmission is enabled. It can be disabled by setting bit DAR (Disable Automatic Retransmission) in CAN Control Register. Further details to this mode are provided in Section 21.8.3.

21.9.2 Auto-Bus-On

Per default, after the DCAN has entered Bus-Off state, the CPU can start a Bus-Off-Recovery sequence by resetting Init bit. If this is not done, the module will stay in Bus-Off state.

The DCAN provides an automatic Auto-Bus-On feature that is enabled by bit ABO in CAN Control Register. If set, the DCAN will automatically start the Bus-Off-Recovery sequence. The sequence can be delayed by a user-defined number of VCLK cycles that can be defined in Auto-Bus-On Time Register.

NOTE: If the DCAN goes Bus-Off due to massive occurrence of CAN bus errors, it stops all bus activities and automatically sets the Init bit. Once the Init bit has been reset by the CPU or due to the Auto-Bus-On feature, the device will wait for 129 occurrences of Bus Idle (equal to 129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus-Off recovery sequence, the error counters will be reset.

21.10 Interrupt Functionality

Interrupts can be generated on two interrupt lines:

1. DCAN0INT line
2. DCAN1INT line

These lines can be enabled by setting the IE0 and IE1 bits, respectively, in the CAN Control Register.

The DCAN provides three groups of interrupt sources: Message Object Interrupts, Status Change Interrupts and Error Interrupts (see Figure 21-13 and Figure 21-14).

The source of an interrupt can be determined by the interrupt identifiers Int0ID / Int1ID in the Interrupt Register (see). When no interrupt is pending, the register will hold the value zero.

Each interrupt line remains active until the dedicated field in the Interrupt Register DCAN INT (Int0ID / Int1ID) again reach zero (this means the cause of the interrupt is reset), or until IE0 / IE1 are reset.

The value 0x8000 in the Int0ID field indicates that an interrupt is pending because the CAN Core has updated (not necessarily changed) the Error and Status Register (Error Interrupt or Status Interrupt). This interrupt has the highest priority. The CPU can update (reset) the status bits WakeUpPnd, RxOk, TxOk and LEC by reading the Error and Status Register DCAN ES, but a write access of the CPU will never generate or reset an interrupt.

Values between 1 and the number of the last message object indicates that the source of the interrupt is one of the message objects, Int0ID resp. Int1ID will point to the pending message interrupt with the highest priority. The Message Object 1 has the highest priority, the last message object has the lowest priority.

An interrupt service routine that reads the message that is the source of the interrupt, may read the message and reset the message object's IntPnd at the same time (ClrIntPnd bit in the IF1/IF2 Command Register). When IntPnd is cleared, the Interrupt Register will point to the next message object with a pending interrupt.

21.10.1 Message Object Interrupts

Message Object interrupts are generated by events from the message objects. They are controlled by the flags IntPND, TxIE and RxIE, that are described in Section 21.5.1.

Message Object interrupts can be routed to either DCAN0INT or DCAN1INT line, controlled by the Interrupt Multiplexer Register (see).

21.10.2 Status Change Interrupts

The events WakeUpPnd, RxOk, TxOk and LEC in Error and Status Register (DCAN ES) belong to the Status Change Interrupts. The Status Change Interrupt group can be enabled by bit in CAN Control Register.

If SIE is set, a Status Change Interrupt will be generated at each CAN frame, independent of bus errors or valid CAN communication, and also independent of the Message RAM configuration.

Status Change interrupts can only be routed to interrupt line DCAN0INT that has to be enabled by setting IE0 in the CAN Control Register.

NOTE: Reading the Error and Status Register will clear the WakeUpPnd flag. If in global power down mode, the WakeUpPnd flag is cleared by such a read access before the DCAN module has been waken up by the system, the DCAN may re-assert the WakeUpPnd flag, and a second interrupt may occur (additional information can be found in Section 21.11.2).

21.10.3 Error Interrupts

The events PER, BOff and EWarn (monitored in Error and Status Register, DCAN ES) belong to the Error Interrupts. The Error Interrupt group can be enabled by setting bit EIE in CAN Control Register.

Error interrupts can only be routed to interrupt line DCAN0INT that has to be enabled by setting IE0 in the CAN Control Register.

Figure 21-13. CAN Interrupt Topology 1

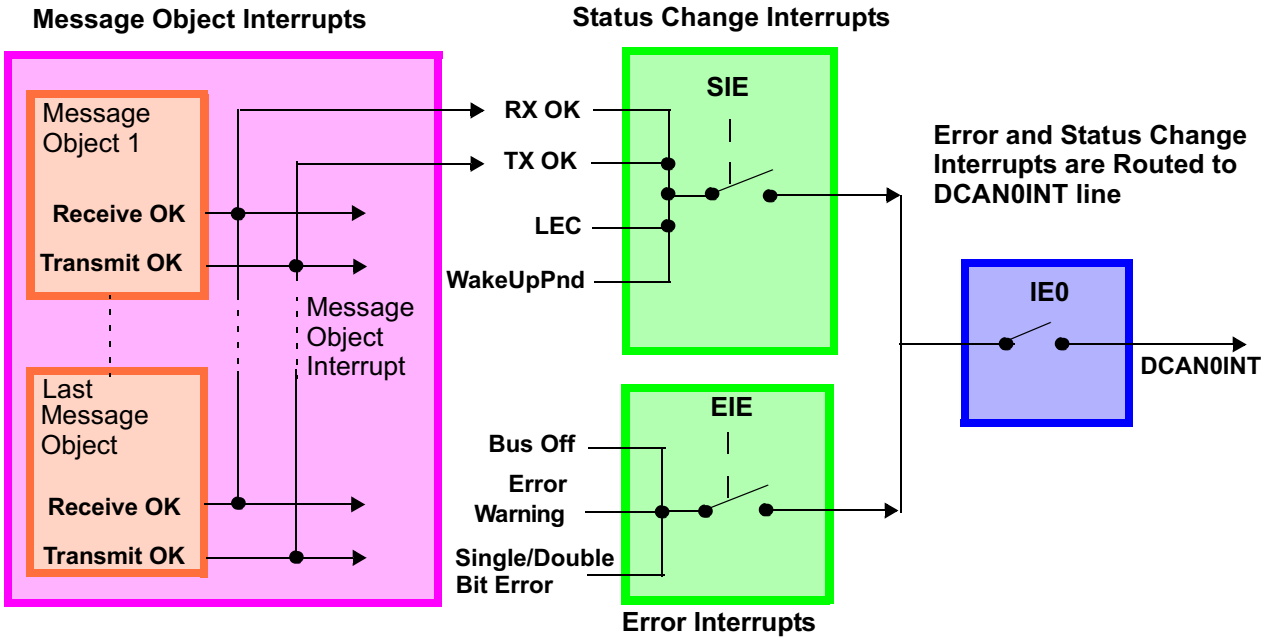
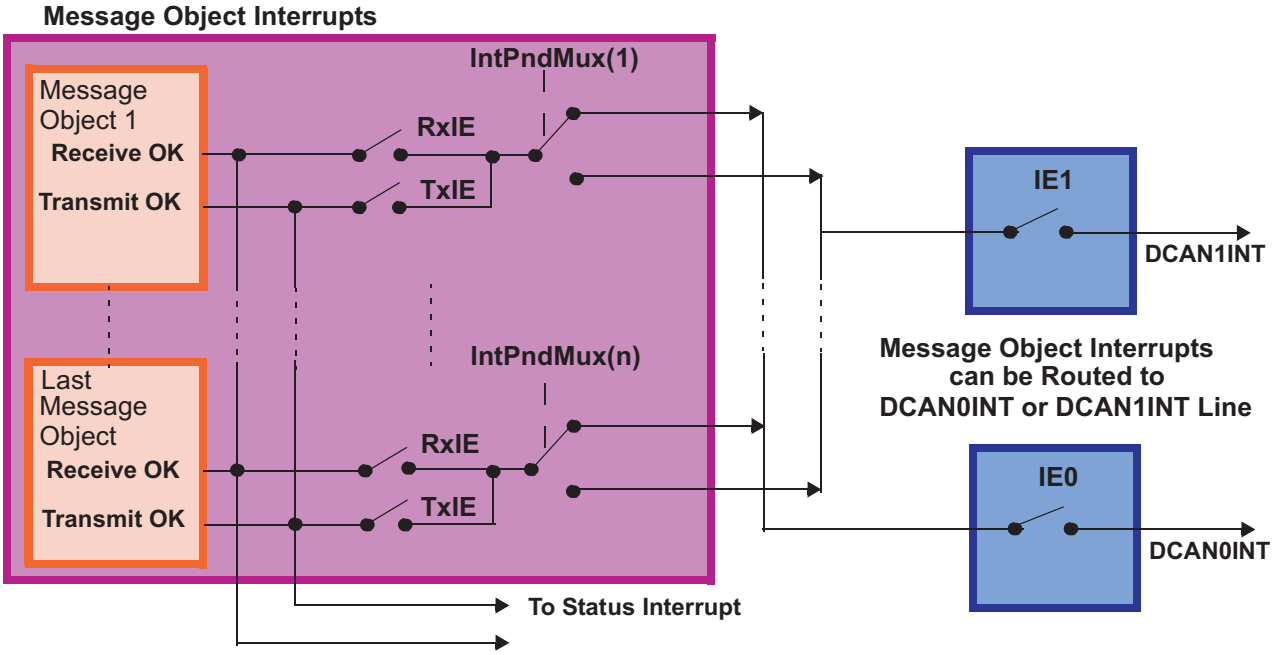


Figure 21-14. CAN Interrupt Topology 2



Details of Interrupt Mapping for actual device will be described in the device specific data sheet.

21.11 Global Power Down Mode

The device architecture supports a centralized global power down control over the peripheral modules through the Peripheral Central Resource (PCR) module (Additional information can be found in Platform Architecture Specification).

21.11.1 Entering Global Power Down Mode

The global power down mode for the DCAN is requested by setting the appropriate Peripheral Power Down Set bit (**PSPWRDWNSETx**) in the PCR module.

The DCAN then finishes all transmit requests of the message objects. When all requests are done, the DCAN waits until a bus idle state is recognized. Then it will automatically set the Initbit to indicate that the global power down mode has been entered.

21.11.2 Wakeup From Global Power Down Mode

When the DCAN module is in global power down mode, a CAN bus activity detection circuit exists, which can be active, if enabled. If this circuit is active,on occurrence of a dominant CAN bus level, the DCAN will set the **WakeUpPnd** bit in Error and Status Register (DCAN ES).

If Status Interrupts are enabled, also an interrupt will be generated. This interrupt could be used by the application to wakeup the DCAN. For this, the application needs to set the appropriate Peripheral Power Down Clear bit (**PSPWRDWNCLRx**) in the PCR module, and to clear the Init bit in CAN Control Register.

After the Init bit has been cleared, the DCAN module waits until it detects 11 consecutive recessive bits on the CAN_RX pin and then goes Bus-Active again.

NOTE: The CAN transceiver circuit has to stay active during CAN bus activity detection. The first CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down mode is lost.

21.12 Local Power Down Mode

Besides the centralized power down mechanism controlled by the PCR module (global power down, see Section 21.15), the DCAN supports a local power down mode that can be controlled within the DCAN control registers.

21.12.1 Entering Local Power Down Mode

The local power down mode is requested by setting the PDR bit in CAN Control Register.

The DCAN then finishes all transmit requests of the message objects. When all requests are done, DCAN waits until a bus idle state is recognized. Then it will automatically set the Initbit in CAN Control Register to prevent any further CAN transfers, and it will also set the PDA bit in CAN Error and Status Register. With setting the PDA bits, the DCAN module indicates that the local power down mode has been entered.

During local power down mode, the internal clocks of the DCAN module are turned off, but there is a wake up logic (see Section 21.12.2) that can be active, if enabled. Also the actual contents of the control registers can be read back.

NOTE: In local low power mode, the application should not clear the Init bit while PDR is set. If there are any messages in the Message RAM configured as to be transmitted and the application resets the init bit, these messages may be sent.

21.12.2 Wakeup From Local Power Down

There are two ways to wake up the DCAN from local power down mode:

1. The application could wake up the DCAN module manually by clearing the PDR bit and then clearing the Init bit in CAN Control Register.
2. Alternatively, a CAN bus activity detection circuit can be activated by setting the wake up on bus activity bit (WUBA) in CAN Control Register. If this circuit is active, on occurrence of a dominant CAN bus level, the DCAN will automatically start the wake up sequence. It will clear the PDR bit in CAN Control Register and also clear the PDA bit in Error and Status Register. The WakeUpPnd bit in CAN Error and Status Register will be set. If Status Interrupts are enabled, also an interrupt will be generated. Finally the Init bit in CAN control register will be cleared.

After the Init bit has been cleared, the module waits until it detects 11 consecutive recessive bits on the CAN_RX pin and then goes Bus-Active again.

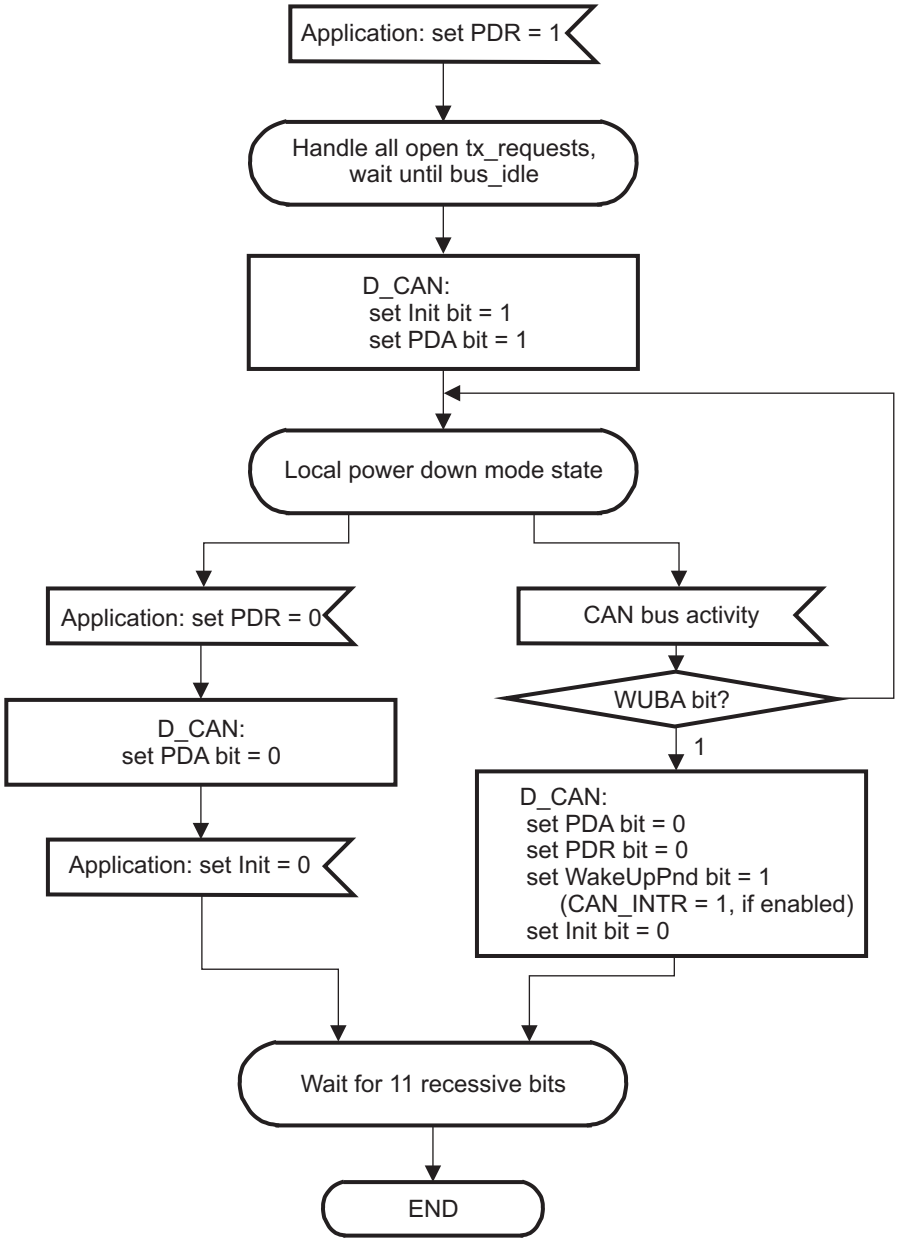
NOTE: The CAN transceiver circuit has to stay active while CAN bus observation. The first CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down and automatic wake-up mode, is lost.

Figure 21-15 shows a flow diagram about entering and leaving local power down mode.

21.13 GIO Support

The CAN_RX and CAN_TX pins of each DCAN module can be used as general purpose IO pins, if CAN functionality is not needed. This function is controlled by the CAN TX IO Control register (see) and the CAN RX IO Control register (see).

Figure 21-15. Local Power Down Mode Flow Diagram



21.14 Test Modes

The DCAN provides several test modes that are mainly intended for production tests or self test.

For all test modes, Test bit in the CAN Control Register needs to be set to one. This enables write access to the Test Register.

NOTE: When using any of the Loop Back modes, it must be ensured by software that all message transfers are finished before setting the Init bit to '1'.

21.14.1 Silent Mode

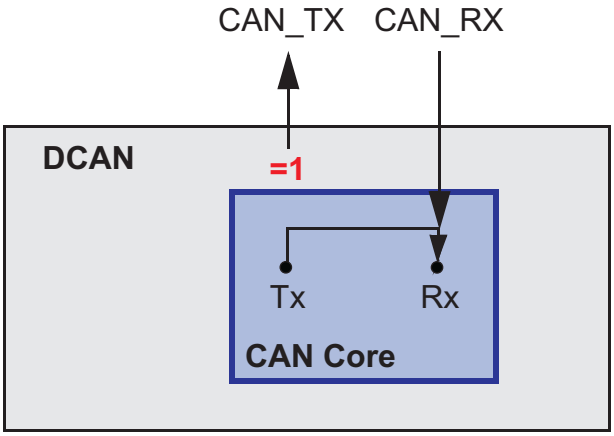
The Silent Mode may be used to analyze the traffic on the CAN bus without affecting it by sending dominant bits (for example, acknowledge bit, overload flag, active error flag). The DCAN is still able to receive valid data frames and valid remote frames, but it will not send any dominant bits. However, these are internally routed to the CAN Core.

Figure 21-16 shows the connection of signals CAN_TX and CAN_RX to the CAN Core in Silent Mode.

Silent Mode can be activated by setting the Silent bit in Test Register to 1.

In ISO 11898-1, the Silent Mode is called the Bus Monitoring Mode.

Figure 21-16. CAN Core in Silent Mode



21.14.2 Loop Back Mode

The Loop Back Mode is mainly intended for hardware self-test functions. In this mode, the CAN Core uses internal feedback from Tx output to Rx input. Transmitted messages are treated as received messages, and can be stored into message objects if they pass acceptance filtering. The actual value of the CAN_RX input pin is disregarded by the CAN Core. Transmitted messages still can be monitored at the CAN_TX pin.

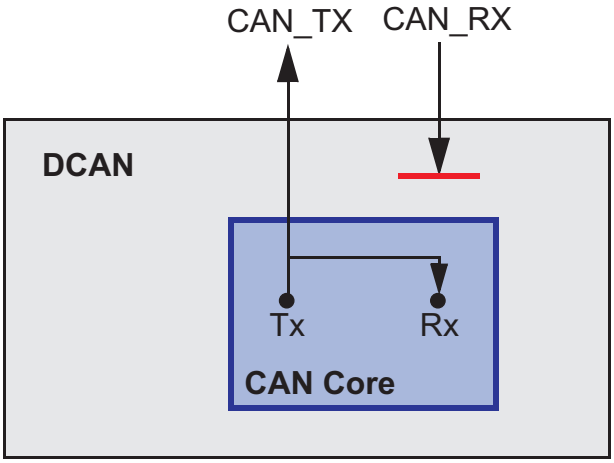
In order to be independent from external stimulation, the CAN Core ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode.

Figure 21-17 shows the connection of signals CAN_TX and CAN_RX to the CAN Core in Loop Back Mode.

Loop Back Mode can be activated by setting bit LBack in Test Register to 1.

NOTE: In Loop Back mode, the signal path from CAN Core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN Core are disregarded. For including these into the testing, see External Loop Back mode (Section 21.14.3).

Figure 21-17. CAN Core in Loop Back Mode



21.14.3 External Loop Back Mode

The External Loop Back Mode is similar to the Loop Back Mode, however it includes the signal path from CAN Core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN Core. When External Loop Back Mode is selected, the input of the CAN core is connected to the input buffer of the Tx pin.

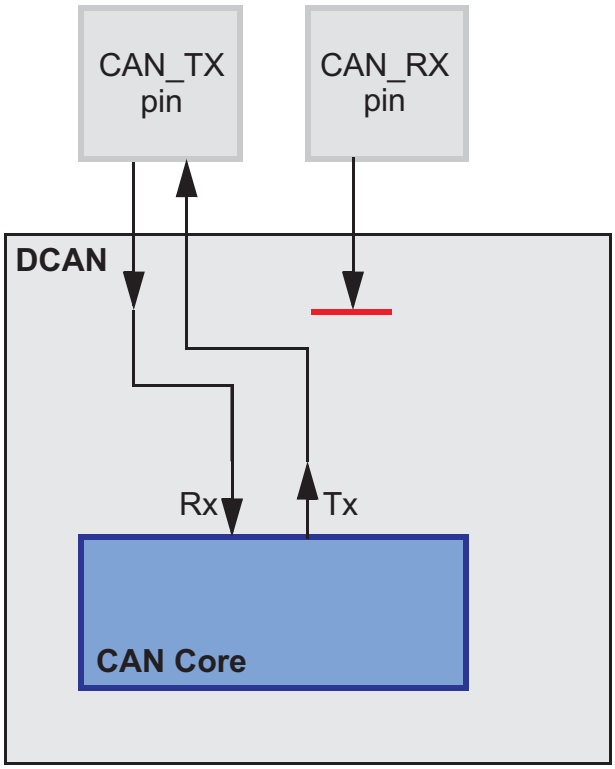
With this configuration, the Tx pin IO circuit can be tested.

External Loop Back Mode can be activated by setting bit ExL in Test Register to 1.

Figure 21-18 shows the connection of signals CAN_TX and CAN_RX to the CAN Core in External Loop Back Mode.

NOTE: When Loop Back Mode is active (LBack bit set), the ExL bit will be ignored.

Figure 21-18. CAN Core in External Loop Back Mode

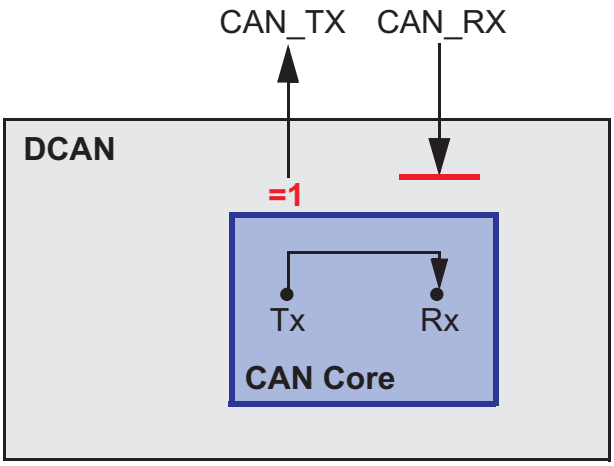


21.14.4 Loop Back Combined with Silent Mode

It is also possible to combine Loop Back Mode and Silent Mode by setting bits LBack and Silent at the same time. This mode can be used for a “Hot Selftest”, that is, the DCAN hardware can be tested without affecting the CAN network. In this mode, the CAN_RX pin is disconnected from the CAN Core and no dominant bits will be sent on the CAN_TX pin.

Figure 21-19 shows the connection of the signals CAN_TX and CAN_RX to the CAN Core in case of the combination of Loop Back Mode with Silent Mode.

Figure 21-19. CAN Core in Loop Back Combined with Silent Mode



21.14.5 Software Control of CAN_TX Pin

Four output functions are available for the CAN transmit pin CAN_TX. Additionally to its default function (serial data output), the CAN_TX pin can drive constant dominant or recessive values, or it can drive the CAN Sample Point signal to monitor the CAN Core’s bit timing.

Combined with the readable value of the CAN_RX pin, this can be used to check the physical layer of the CAN bus.

The output mode of pin CAN_TX is selected by programming the Test Register bits Tx[1:0] as described in .

NOTE: The software control for pin CAN_TX interferes with CAN protocol functions. For CAN message transfer or any of the test modes Loop Back Mode, External Loop Back Mode or Silent Mode, the CAN_TX pin should operate in its default functionality.

21.15 SECDED Mechanism

The DCAN module provides a single error correction and double error detection (SECDED) mechanism to ensure data integrity of Message RAM data. For each message object (136 bits) in the Message RAM, 9 ECC bits will be calculated. See Section 21.5.5.

The ECC bits are stored in a dedicated RAM. They will be generated on write accesses and will be checked on read accesses.

The SECDED functionality can be enabled or disabled by PMD bit field in CAN Control Register. If SECDED is enabled, ECC bits will be automatically generated and checked.

With the ECCMODE field in the ECC Control and Status register the single bit error correction can be enabled or disabled (default: enabled).

NOTE: During RAM initialization, no ECC check will be done, but if the PMD bit is set, the ECC bits will be generated.

21.15.1 Behavior on Single Bit Error

If a single bit error is detected with single bit error correction enabled, the correction will be done and the SEFLG in the ECC Control and Status register will be set.

If single bit error correction is disabled and a single bit error is detected then the SEFLG in the ECC Control and Status register and the the PER bit in the Error and Status register will be set. If error interrupts are enabled, also an interrupt would be generated. In order to avoid the transmission of invalid data over the CAN bus, the MsgVal bit of the message object will be reset.

The message object number where the single bit error has occurred will be indicated in the ECC Single Bit Error Code Register.

When single bit error correction is disabled the message object data can be read by the host CPU, independently of single bit errors. Thus, the application has to ensure that the read data is valid, for example, by immediately checking the ECC Single Bit Error Code Register on single bit error interrupt.

21.15.2 Behavior on Double Bit Error

If a double bit error is detected, then the DEFLG in the ECC Control and Status register and the PER bit in Error and Status Register will be set. If error interrupts are enabled, also an interrupt would be generated. In order to avoid the transmission of invalid data over the CAN bus, the MsgVal bit of the message object will be reset. The message object number will be indicated in the Parity Error Code Register.

The message object data can be read by the host CPU, independently of double bit errors. Thus, the application has to ensure that the read data is valid, for example, by immediately checking the Parity Error Code register on double bit error interrupt.

21.15.3 SECDED Testing

Testing of the SECDED mechanism can be implemented by using the diagnostic mode, which is enabled with the ECCDIAG register. The following procedure can be used:

1. Disable SECDED using DCAN control register. Enable diagnostic mode using the ECCDIAG register

2. Write to corrupt the data (in RDA mode) or ECC bits.

3. Enable SECDED and read data for which ECC is corrupted (either in RDA mode or via IFx registers).

4. Single bit error or double bit error flag will be set in the diagnostic status register (ECCDIAG STAT) and in the ECC Control and Status register accordingly. A double bit error or a single bit error with single bit error correction disabled also triggers the PER flag.

5. Disable diagnostic mode.

21.16 Debug/Suspend Mode

The module supports the usage of an external debug unit by providing functions like pausing DCAN activities and making Message RAM content accessible via VBUSP interface.

Before entering debug/suspend mode, the circuit will either wait until a started transmission or reception will be finished and Bus idle state is recognized, or immediately interrupt a current transmission or reception. This is depending on bit IDS in CAN Control Register.

Afterwards, the DCAN enters debug/suspend mode, indicated by InitDbg flag in CAN Control Register.

During Debug/Suspend mode, all DCAN registers can be accessed. Reading reserved bits will return 0; writing to reserved bits will have no effect.

Also, the Message RAM will be memory mapped. This allows the external debug unit to read the Message RAM. For the memory organization, see Section 21.5.3).

NOTE: During Debug/Suspend Mode, the Message RAM cannot be accessed via the IFx register sets.

Writing to control registers in debug/suspend mode may influence the CAN state machine and further message handling.

For debug support, the auto clear functionality of the following DCAN registers is disabled:

- Error and Status Register (clear of status flags by read)
- IF1/IF2 Command Registers (clear of DMAActive flag by read/write)

21.17 MSS_DCAN Registers

Table 21-6 lists the memory-mapped registers for the MSS_DCAN. All register offset addresses not listed in Table 21-6 should be considered as reserved locations and the register contents should not be modified.

Table 21-6. MSS_DCAN Registers

Offset	Acronym	Register Name	Section
0h	CTL	CAN Control Register	Section 21.17.1
4h	ES	Error and Status Register	Section 21.17.2
8h	ERRC	Error Counter Register	Section 21.17.3
Ch	BTR	Bit Timing Register	Section 21.17.4
10h	INT	Interrupt Register	Section 21.17.5
14h	TEST	Test Register	Section 21.17.6
18h	Reserved_1		Section 21.17.7
1Ch	PERR	Parity Error Code Register	Section 21.17.8
20h	DCAN_REV_ID	DCAN Revision ID	Section 21.17.9
24h	ECCDIAG	ECC Diagnostic Register	Section 21.17.10
28h	ECCDIAG_STAT	ECC Diagnostic Status Register	Section 21.17.11
2Ch	ECC_CS	ECC Control and Status Register	Section 21.17.12
30h	ECC_SERR	ECC Single Bit Error Code Register	Section 21.17.13
34h	TTCAN1	TTCAN1	Section 21.17.14
38h	TTCAN2	TTCAN2	Section 21.17.15
3Ch	TTCAN3	TTCAN3	Section 21.17.16
40h	TTCAN4	TTCAN4	Section 21.17.17
44h	TTCAN5	TTCAN5	Section 21.17.18
48h	TTCAN6	TTCAN6	Section 21.17.19
4Ch	TTCAN7	TTCAN7	Section 21.17.20
50h	TTCAN8	TTCAN8	Section 21.17.21
54h	TTCAN9	TTCAN9	Section 21.17.22
58h	TTCAN10	TTCAN10	Section 21.17.23
5Ch	TTCAN11	TTCAN11	Section 21.17.24
60h	TTCAN12	TTCAN12	Section 21.17.25
64h	TTCAN13	TTCAN13	Section 21.17.26
68h	TTCAN14	TTCAN14	Section 21.17.27
6Ch	TTCAN15	TTCAN15	Section 21.17.28
70h	TTCAN16	TTCAN16	Section 21.17.29
74h	TTCAN17	TTCAN17	Section 21.17.30
78h	TTCAN18	TTCAN18	Section 21.17.31
7Ch	TTCAN19	TTCAN19	Section 21.17.32
80h	ABOTR	Auto-Bus-On Time Register	Section 21.17.33
84h	TXRQ_X	Transmission Request X Register	Section 21.17.34
88h	TXRQ12	Transmission Request Register 12	Section 21.17.35
8Ch	TXRQ34	Transmission Request Register 34	Section 21.17.36
90h	TXRQ56	Transmission Request Register 56	Section 21.17.37
94h	TXRQ78	Transmission Request Register 78	Section 21.17.38
98h	NWDAT_X	New Data X Register	Section 21.17.39
9Ch	NWDAT12	New Data Register 12	Section 21.17.40
A0h	NWDAT34	New Data Register 34	Section 21.17.41
A4h	NWDAT56	New Data Register 56	Section 21.17.42
A8h	NWDAT78	New Data Register 78	Section 21.17.43

Table 21-6. MSS_DCAN Registers (continued)

Offset	Acronym	Register Name	Section
ACh	INTPND_X	Interrupt Pending X Register	Section 21.17.44
B0h	INTPND12	Interrupt Pending Register 12	Section 21.17.45
B4h	INTPND34	Interrupt Pending Register 34	Section 21.17.46
B8h	INTPND56	Interrupt Pending Register 56	Section 21.17.47
BCh	INTPND78	Interrupt Pending Register 78	Section 21.17.48
C0h	MSGVAL_X	Message Valid X Register	Section 21.17.49
C4h	MSGVAL12	Message Valid Register 12	Section 21.17.50
C8h	MSGVAL34	Message Valid Register 34	Section 21.17.51
CCh	MSGVAL56	Message Valid Register 56	Section 21.17.52
D0h	MSGVAL78	Message Valid Register 78	Section 21.17.53
D4h	Reserved_2		Section 21.17.54
D8h	INTMUX12	Interrupt Multiplexer Register 12	Section 21.17.55
DCh	INTMUX34	Interrupt Multiplexer Register 34	Section 21.17.56
E0h	INTMUX56	Interrupt Multiplexer Register 56	Section 21.17.57
E4h	INTMUX78	Interrupt Multiplexer Register 78	Section 21.17.58
E8h	Reserved_3		Section 21.17.59
ECh	Reserved_4		Section 21.17.60
F0h	Reserved_5		Section 21.17.61
F4h	Reserved_6		Section 21.17.62
F8h	Reserved_7		Section 21.17.63
FCh	Reserved_8		Section 21.17.64
100h	IF1CMD	IF1 Command Registers	Section 21.17.65
104h	IF1MSK	IF1 Mask Register	Section 21.17.66
108h	IF1ARB	IF1 Arbitration Register	Section 21.17.67
10Ch	IF1MCTL	IF1 Message Control Register	Section 21.17.68
110h	IF1DATA	IF1 Data A Register	Section 21.17.69
114h	IF1DATB	IF1 Data B Register	Section 21.17.70
118h	Reserved_9		Section 21.17.71
11Ch	Reserved_10		Section 21.17.72
120h	IF2CMD	IF2 Command Registers	Section 21.17.73
124h	IF2MSK	IF2 Mask Register	Section 21.17.74
128h	IF2ARB	IF2 Arbitration Register	Section 21.17.75
12Ch	IF2MCTL	IF2 Message Control Register	Section 21.17.76
130h	IF2DATA	IF2 Data A Register	Section 21.17.77
134h	IF2DATB	IF2 Data B Register	Section 21.17.78
138h	Reserved_11		Section 21.17.79
13Ch	Reserved_12		Section 21.17.80
140h	IF3OBS	IF3 Observation Register	Section 21.17.81
144h	IF3MSK	IF3 Mask Register	Section 21.17.82
148h	IF3ARB	IF3 Arbitration Register	Section 21.17.83
14Ch	IF3MCTL	IF3 Message Control Register	Section 21.17.84
150h	IF3DATA	IF3 Data A Register	Section 21.17.85
154h	IF3DATB	IF3 Data B Register	Section 21.17.86
158h	Reserved_13		Section 21.17.87
15Ch	Reserved_14		Section 21.17.88
160h	IF3UPD12	IF3 Update Enable Register 12	Section 21.17.89
164h	IF3UPD34	IF3 Update Enable Register 34	Section 21.17.90

Table 21-6. MSS_DCAN Registers (continued)

Offset	Acronym	Register Name	Section
168h	IF3UPD56	IF3 Update Enable Register 56	Section 21.17.91
16Ch	IF3UPD78	IF3 Update Enable Register 78	Section 21.17.92
170h	Reserved_15		Section 21.17.93
174h	Reserved_16		Section 21.17.94
178h	Reserved_17		Section 21.17.95
17Ch	Reserved_18		Section 21.17.96
180h	Reserved_19		Section 21.17.97
184h	Reserved_20		Section 21.17.98
188h	Reserved_21		Section 21.17.99
18Ch	Reserved_22		Section 21.17.100
190h	Reserved_23		Section 21.17.101
194h	Reserved_24		Section 21.17.102
198h	Reserved_25		Section 21.17.103
19Ch	Reserved_26		Section 21.17.104
1A0h	Reserved_27		Section 21.17.105
1A4h	Reserved_28		Section 21.17.106
1A8h	Reserved_29		Section 21.17.107
1ACh	Reserved_30		Section 21.17.108
1B0h	Reserved_31		Section 21.17.109
1B4h	Reserved_32		Section 21.17.110
1B8h	Reserved_33		Section 21.17.111
1BCh	Reserved_34		Section 21.17.112
1C0h	Reserved_35		Section 21.17.113
1C4h	Reserved_36		Section 21.17.114
1C8h	Reserved_37		Section 21.17.115
1CCh	Reserved_38		Section 21.17.116
1D0h	Reserved_39		Section 21.17.117
1D4h	Reserved_40		Section 21.17.118
1D8h	Reserved_41		Section 21.17.119
1DCh	Reserved_42		Section 21.17.120
1E0h	TIOC	CAN TX IO Control Register	Section 21.17.121
1E4h	RIOC	CAN RX IO Control Register	Section 21.17.122

Complex bit access types are encoded to fit into small table cells. Table 21-7 shows the codes that are used for access types in this section.

Table 21-7. MSS_DCAN Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WP	P W	Requires privileged access Write
Reset or Default Value		
-n		Value after reset or the default value

21.17.1 CTL Register (Offset = 0h) [reset = 1401h]

CTL is shown in Figure 21-20 and described in Table 21-8.

Return to Summary Table.

CAN Control Register The Bus-Off recovery sequence (refer to CAN specification) cannot be shortened by setting or resetting Init bit. If the module goes Bus-Off, it will automatically set the Init bit and stop all bus activities. When the Init bit is cleared by the application again, the module will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the bus-off recovery sequence, the error counters will be reset. After the Init bit is reset, each time when a sequence of 11 recessive bits is monitored, a Bit0 error code is written to the error and status register, enabling the CPU to check whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the proceeding of the bus-off recovery sequence.

Figure 21-20. CTL Register

31	30	29	28	27	26	25	24
RESERVED						WUBA	PDR
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED			DE3	DE2	DE1	IE1	InitDbg
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
SWR	RESERVED	PMD				ABO	IDS
R/WP-0h	R-0h	R/W-5h				R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
Test	CCE	DAR	RESERVED	EIE	SIE	IE0	Init
R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 21-8. CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	WUBA	R/W	0h	Automatic wake up on bus activity when in local power-down mode. Note: The CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down and automatic wake-up mode, will be lost. 0h (R/W) = No detection of a dominant CAN bus level while in local power-down mode. 1h (R/W) = Detection of a dominant CAN bus level while in local power-down mode is enabled. On occurrence of a dominant CAN bus level, the wake up sequence is started.
24	PDR	R/W	0h	Request for local low power-down mode 0h (R/W) = No application request for local low power-down mode. If the application has cleared this bit while DCAN in local power-down mode, also the Init bit has to be cleared. 1h (R/W) = Local power-down mode has been requested by application. The DCAN will acknowledge the local power-down mode by setting bit PDA in the error and status register. The local clocks will be turned off by DCAN internal logic.
23-21	RESERVED	R	0h	
20	DE3	R/W	0h	Enable DMA request line for IF3. Note: A pending DMA request for IF3 remains active until first access to one of the IF3 registers. 0h (R/W) = Disabled 1h (R/W) = Enabled
19	DE2	R/W	0h	Enable DMA request line for IF2. Note: A pending DMA request for IF2 remains active until first access to one of the IF2 registers. 0h (R/W) = Disabled 1h (R/W) = Enabled

Table 21-8. CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	DE1	R/W	0h	Enable DMA request line for IF1. Note: A pending DMA request for IF1 remains active until first access to one of the IF1 registers. 0h (R/W) = Disabled 1h (R/W) = Enabled
17	IE1	R/W	0h	Interrupt line 1 enable 0h (R/W) = Disabled - Module interrupt DCAN1INT is always low. 1h (R/W) = Enabled - interrupts will assert line DCAN1INT to one; line remains active until pending interrupts are processed.
16	InitDbg	R	0h	Internal init state while debug access 0h (R/W) = Not in debug mode, or debug mode requested but not entered. 1h (R/W) = Debug mode requested and internally entered; the DCAN is ready for debug accesses.
15	SWR	R/WP	0h	SW reset enable. Note: To execute software reset, the following procedure is necessary: (a) Set Init bit to shut down CAN communication and (b) Set SWR bit additionally to Init bit. 0h (R/W) = Normal Operation 1h (R/W) = Module is forced to reset state. This bit will automatically get cleared after execution of SW reset after one OCP clock cycle.
14	RESERVED	R	0h	
13-10	PMD	R/W	5h	Parity on/off. 5 = Parity function disabled. Others = Parity function enabled.
9	ABO	R/W	0h	Auto-Bus-On enable 0h (R/W) = The Auto-Bus-On feature is disabled 1h (R/W) = The Auto-Bus-On feature is enabled
8	IDS	R/W	0h	Interruption debug support enable 0h (R/W) = When Debug/Suspend mode is requested, DCAN will wait for a started transmission or reception to be completed before entering Debug/Suspend mode 1h (R/W) = When Debug/Suspend mode is requested, DCAN will interrupt any transmission or reception, and enter Debug/Suspend mode immediately.
7	Test	R/W	0h	Test mode enable 0h (R/W) = Normal Operation 1h (R/W) = Test Mode
6	CCE	R/W	0h	Configuration change enable 0h (R/W) = The CPU has no write access to the configuration registers. 1h (R/W) = The CPU has write access to the configuration registers (when Init bit is set).
5	DAR	R/W	0h	Disable automatic retransmission 0h (R/W) = Automatic retransmission of not successful messages enabled. 1h (R/W) = Automatic retransmission disabled.
4	RESERVED	R	0h	
3	EIE	R/W	0h	Error interrupt enable 0h (R/W) = Disabled - PER, BOff and EWarn bits can not generate an interrupt. 1h (R/W) = Enabled - PER, BOff and EWarn bits can generate an interrupt at DCAN0INT line and affect the interrupt register.

Table 21-8. CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	SIE	R/W	0h	Status change interrupt enable 0h (R/W) = Disabled - WakeUpPnd, RxOk, TxOk and LEC bits can not generate an interrupt. 1h (R/W) = Enabled - WakeUpPnd, RxOk, TxOk and LEC can generate an interrupt at DCAN0INT line and affect the interrupt register.
1	IE0	R/W	0h	Interrupt line 0 enable 0h (R/W) = Disabled - Module interrupt DCAN0INT is always low. 1h (R/W) = Enabled - interrupts will assert line DCAN0INT to one; line remains active until pending interrupts are processed.
0	Init	R/W	1h	Initialization 0h (R/W) = Normal operation 1h (R/W) = Initialization mode is entered

21.17.2 ES Register (Offset = 4h) [reset = 7h]

ES is shown in Figure 21-21 and described in Table 21-9.

Return to Summary Table.

Status Register Interrupts are generated by bits PER, BOff and EWarn (if EIE bit in CAN control register is set) and by bits WakeUpPnd, RxOk, TxOk, and LEC (if SIE bit in CAN control register is set). A change of bit EPass will not generate an interrupt. Reading the error and status register clears the WakeUpPnd, PER, RxOk and TxOk bits and set the LEC to value '7.' Additionally, the status interrupt value (0x8000) in the interrupt register will be replaced by the next lower priority interrupt value. The EOI for all other interrupts (DCANINT0 and DCANINT1) are automatically handled by hardware. For debug support, the auto clear functionality of error and status register (clear of status flags by read) is disabled when in debug/suspend mode.

Figure 21-21. ES Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					PDA	WakeUp_Pnd	PER_
R-0h					R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
BOff	EWarn	EPass	RxOk	TxOk	LEC		
R-0h	R-0h	R-0h	R-0h	R-0h	R-7h		

Table 21-9. ES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	PDA	R	0h	Local power-down mode acknowledge 0h (R/W) = DCAN is not in local power-down mode. 1h (R/W) = Application request for setting DCAN to local power-down mode was successful. DCAN is in local power-down mode.
9	WakeUp_Pnd	R	0h	Wake up pending. This bit can be used by the CPU to identify the DCAN as the source to wake up the system. This bit will be reset if error and status register is read. 0h (R/W) = No Wake Up is requested by DCAN. 1h (R/W) = DCAN has initiated a wake up of the system due to dominant CAN bus while module power down.
8	PER_	R	0h	Parity error detected. This bit will be reset if error and status register is read. 0h (R/W) = No parity error has been detected since last read access. 1h (R/W) = The parity check mechanism has detected a parity error in the Message RAM.
7	BOff	R	0h	Bus-Off state 0h (R/W) = The CAN module is not bus-off state. 1h (R/W) = The CAN module is in bus-off state.
6	EWarn	R	0h	Warning state 0h (R/W) = Both error counters are below the error warning limit of 96. 1h (R/W) = At least one of the error counters has reached the error warning limit of 96.

Table 21-9. ES Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	EPass	R	0h	Error passive state 0h (R/W) = On CAN Bus error, the DCAN could send active error frames. 1h (R/W) = The CAN core is in the error passive state as defined in the CAN Specification.
4	RxOk	R	0h	Received a message successfully. This bit will be reset if error and status register is read. 0h (R/W) = No message has been successfully received since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events. 1h (R/W) = A message has been successfully received since the last time when this bit was reset by a read access of the CPU (independent of the result of acceptance filtering).
3	TxOk	R	0h	Transmitted a message successfully. This bit will be reset if error and status register is read. 0h (R/W) = No message has been successfully transmitted since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events. 1h (R/W) = A message has been successfully transmitted (error free and acknowledged by at least one other node) since the last time when this bit was reset by a read access of the CPU.
2-0	LEC	R	7h	Last error code. The LEC field indicates the type of the last error on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. 0h (R/W) = No error 1h (R/W) = Stuff error. More than five equal bits in a row have been detected in a part of a received message where this is not allowed. 2h (R/W) = Form error. A fixed format part of a received frame has the wrong format. 3h (R/W) = Ack error. The message this CAN core transmitted was not acknowledged by another node. 4h (R/W) = Bit1 error. During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 5h (R/W) = Bit0 error. During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (logical value '0'), but the monitored bus level was recessive. During Bus-Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus-Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 6h (R/W) = CRC error. In a received message, the CRC check sum was incorrect. (CRC received for an incoming message does not match the calculated CRC for the received data). 7h (R/W) = No CAN bus event was detected since the last time the CPU read the error and status register. Any read access to the error and status register re-initializes the LEC to value '7.'

21.17.3 ERRC Register (Offset = 8h) [reset = 0h]

ERRC is shown in Figure 21-22 and described in Table 21-10.

Return to Summary Table.

Error Counter Register

Figure 21-22. ERRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC							TEC							
R-0h	R-0h							R-0h							

Table 21-10. ERRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	RP	R	0h	Receive error passive 0h (R/W) = The receive error counter is below the error passive level. 1h (R/W) = The receive error counter has reached the error passive level as defined in the CAN specification.
14-8	REC	R	0h	Receive error counter. Actual state of the receive error counter (values from 0 to 255).
7-0	TEC	R	0h	Transmit error counter. Actual state of the transmit error counter (values from 0 to 255).

21.17.4 BTR Register (Offset = Ch) [reset = 2301h]

BTR is shown in Figure 21-23 and described in Table 21-11.

Return to Summary Table.

Bit Timing/BRP Extension Register This register is only writable if CCE and Init bits in the CAN control register are set. The CAN bit time may be programmed in the range of 8 to 25 time quanta. The CAN time quantum may be programmed in the range of 1 to1024 CAN_CLK periods. With a CAN_CLK of 8 MHz and BRPE = 0x00, the reset value of 0x00002301 configures the DCAN for a bit rate of 500kBit/s.

Figure 21-23. BTR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				BRPE			
R-0h				R-0h			
15	14	13	12	11	10	9	8
RESERVED	TSeg2			TSeg1			
R-0h	R/WP-2h			R/WP-3h			
7	6	5	4	3	2	1	0
SJW		BRP					
R/WP-0h		R/WP-1h					

Table 21-11. BTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-16	BRPE	R	0h	Baud rate prescaler extension. Valid programmed values are 0 to 15. By programming BRPE the baud rate prescaler can be extended to values up to 1024.
15	RESERVED	R	0h	
14-12	TSeg2	R/WP	2h	Time segment after the sample point. Valid programmed values are 0 to 7. The actual TSeg2 value which is interpreted for the bit timing will be the programmed TSeg2 value + 1.
11-8	TSeg1	R/WP	3h	Time segment before the sample point. Valid programmed values are 1 to 15. The actual TSeg1 value interpreted for the bit timing will be the programmed TSeg1 value + 1.
7-6	SJW	R/WP	0h	Synchronization Jump Width. Valid programmed values are 0 to 3. The actual SJW value interpreted for the synchronization will be the programmed SJW value + 1.
5-0	BRP	R/WP	1h	Baud rate prescaler. Value by which the CAN_CLK frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid programmed values are 0 to 63. The actual BRP value interpreted for the bit timing will be the programmed BRP value + 1.

21.17.5 INT Register (Offset = 10h) [reset = 0h]

INT is shown in Figure 21-24 and described in Table 21-12.

Return to Summary Table.

Interrupt Register

Figure 21-24. INT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								Int1ID								Int0ID															
R-0h								R-0h								R-0h															

Table 21-12. INT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	Int1ID	R	0h	Interrupt 1 Identifier (indicates the message object with the highest pending interrupt). If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN1INT interrupt line remains active until Int1ID reaches value 0 (the cause of the interrupt is reset) or until IE1 is cleared. A message interrupt is cleared by clearing the message object's IntPnd bit. Among the message interrupts, the message object's interrupt priority decreases with increasing message number. 0h (R/W) = No interrupt is pending. 80h (R/W) = Number of message object which caused the interrupt.
15-0	Int0ID	R	0h	Interrupt Identifier (the number here indicates the source of the interrupt). If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN0INT interrupt line remains active until Int0ID reaches value 0 (the cause of the interrupt is reset) or until IE0 is cleared. The Status interrupt has the highest priority. Among the message interrupts, the message object's interrupt priority decreases with increasing message number. 0h (R/W) = Unused. 0080h (R/W) = Number of message object which caused the interrupt. 1F40h (R/W) = Error and status register value is not 0x07.

21.17.6 TEST Register (Offset = 14h) [reset = 80h]

TEST is shown in Figure 21-25 and described in Table 21-13.

Return to Summary Table.

Test Register For all test modes, the test bit in CAN control register needs to be set to one. If test bit is set, the RDA, EXL, Tx1, Tx0, LBack and Silent bits are writable. Bit Rx monitors the state of pin CAN_RX and therefore is only readable. All test register functions are disabled when test bit is cleared. The test register is only writable if test bit in CAN control register is set. Setting Tx[1:0] other than '00' will disturb message transfer. When the internal loop-back mode is active (bit LBack is set), bit EXL will be ignored.

Figure 21-25. TEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						RDA	EXL
R-0h						R/WP-0h	R/WP-0h
7	6	5	4	3	2	1	0
Rx	Tx		LBack	Silent	RESERVED		
R-1h	R/WP-0h		R/WP-0h	R/WP-0h	R-0h		

Table 21-13. TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	RDA	R/WP	0h	RAM direct access enable 0h (R/W) = Normal operation 1h (R/W) = Direct access to the RAM is enabled while in test mode
8	EXL	R/WP	0h	External loopback mode 0h (R/W) = Disabled 1h (R/W) = Enabled
7	Rx	R	1h	Receive pin. Monitors the actual value of the CAN_RX pin 0h (R/W) = The CAN bus is dominant 1h (R/W) = The CAN bus is recessive
6-5	Tx	R/WP	0h	Control of CAN_TX pin. 0h (R/W) = Normal operation, CAN_TX is controlled by the CAN core. 1h (R/W) = Sample point can be monitored at CAN_TX pin. Ah (R/W) = CAN_TX pin drives a dominant value. Bh (R/W) = CAN_TX pin drives a recessive value.
4	LBack	R/WP	0h	Loopback mode 0h (R/W) = Disabled 1h (R/W) = Enabled
3	Silent	R/WP	0h	Silent mode 0h (R/W) = Disabled 1h (R/W) = Enabled
2-0	RESERVED	R	0h	

21.17.7 Reserved_1 Register (Offset = 18h) [reset = 0h]

Reserved_1 is shown in Figure 21-26 and described in Table 21-14.

Return to Summary Table.

Figure 21-26. Reserved_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-14. Reserved_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.8 PERR Register (Offset = 1Ch) [reset = 0h]

PERR is shown in Figure 21-27 and described in Table 21-15.

Return to Summary Table.

Parity Error Code Register If a parity error is detected, the PER flag will be set in the error and status register. This bit is not reset by the parity check mechanism; it must be reset by reading the error and status register. In addition to the PER flag, the parity error code register will indicate the memory area where the parity error has been detected (message number and word number). If more than one word with a parity error was detected, the highest word number with a parity error will be displayed. After a parity error has been detected, the register will hold the last error code until power is removed.

Figure 21-27. PERR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					Word_Number		
R-0h					R-0h		
7	6	5	4	3	2	1	0
Message_Number							
R-0h							

Table 21-15. PERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-8	Word_Number	R	0h	Word number where parity error has been detected. 05h (R/W) = RDA word number (1 to 5) of the message object (according to the message RAM representation in RDA mode).
7-0	Message_Number	R	0h	Message number. 80h (R/W) = Message object number where parity error has been detected

21.17.9 DCAN_REV_ID Register (Offset = 20h) [reset = 0h]

DCAN_REV_ID is shown in Figure 21-28 and described in Table 21-16.

Return to Summary Table.

DCAN Revision ID

Figure 21-28. DCAN_REV_ID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewBitfield																															
R-																															

Table 21-16. DCAN_REV_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NewBitfield	R	0	

21.17.10 ECCDIAG Register (Offset = 24h) [reset = 0h]

ECCDIAG is shown in Figure 21-29 and described in Table 21-17.

Return to Summary Table.

ECC Diagnostic Register

Figure 21-29. ECCDIAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												ECC_DIAG			
R-												R/W-			

Table 21-17. ECCDIAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R		
3-0	ECC_DIAG	R/W		

21.17.11 ECCDIAG_STAT Register (Offset = 28h) [reset = 0h]

ECCDIAG_STAT is shown in Figure 21-30 and described in Table 21-18.

Return to Summary Table.

ECC Diagnostic Status Register

Figure 21-30. ECCDIAG_STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-							
23	22	21	20	19	18	17	16
RESERVED							
R-							
15	14	13	12	11	10	9	8
RESERVED							DEFLG_DIAG
R-							R/W-
7	6	5	4	3	2	1	0
RESERVED							SEFLG_DIAG
R-							R/W-

Table 21-18. ECCDIAG_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R		
8	DEFLG_DIAG	R/W		
7-1	RESERVED	R		
0	SEFLG_DIAG	R/W		

21.17.12 ECC_CS Register (Offset = 2Ch) [reset = 0h]

ECC_CS is shown in Figure 21-31 and described in Table 21-19.

Return to Summary Table.

ECC Control and Status Register

Figure 21-31. ECC_CS Register

31	30	29	28	27	26	25	24
RESERVED				SBE_EVT_EN			
R-				R/W-			
23	22	21	20	19	18	17	16
RESERVED				ECCMODE			
R-				R/W-			
15	14	13	12	11	10	9	8
RESERVED							DEFLG
R-							R/W-
7	6	5	4	3	2	1	0
RESERVED							SEFLG
R-							R/W-

Table 21-19. ECC_CS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R		
27-24	SBE_EVT_EN	R/W		
23-20	RESERVED	R		
19-16	ECCMODE	R/W		
15-9	RESERVED	R		
8	DEFLG	R/W		
7-1	RESERVED	R		
0	SEFLG	R/W		

21.17.13 ECC_SERR Register (Offset = 30h) [reset = 0h]

ECC_SERR is shown in Figure 21-32 and described in Table 21-20.

Return to Summary Table.

ECC Single Bit Error Code Register

Figure 21-32. ECC_SERR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MSG_NUM							
R-																								R/W-							

Table 21-20. ECC_SERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R		
7-0	MSG_NUM	R/W		

21.17.14 TTCAN1 Register (Offset = 34h) [reset = 0h]

TTCAN1 is shown in Figure 21-33 and described in Table 21-21.

Return to Summary Table.

TTCAN1

Figure 21-33. TTCAN1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-21. TTCAN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.15 TTCAN2 Register (Offset = 38h) [reset = 0h]

TTCAN2 is shown in Figure 21-34 and described in Table 21-22.

Return to Summary Table.

TTCAN2

Figure 21-34. TTCAN2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-22. TTCAN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.16 TTCAN3 Register (Offset = 3Ch) [reset = 0h]

TTCAN3 is shown in Figure 21-35 and described in Table 21-23.

Return to Summary Table.

TTCAN3

Figure 21-35. TTCAN3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-23. TTCAN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.17 TTCAN4 Register (Offset = 40h) [reset = 0h]

TTCAN4 is shown in Figure 21-36 and described in Table 21-24.

Return to Summary Table.

TTCAN4

Figure 21-36. TTCAN4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-24. TTCAN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.18 TTCAN5 Register (Offset = 44h) [reset = 0h]

TTCAN5 is shown in Figure 21-37 and described in Table 21-25.

Return to Summary Table.

TTCAN5

Figure 21-37. TTCAN5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-25. TTCAN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.19 TTCAN6 Register (Offset = 48h) [reset = 0h]

TTCAN6 is shown in Figure 21-38 and described in Table 21-26.

Return to Summary Table.

TTCAN6

Figure 21-38. TTCAN6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-26. TTCAN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.20 TTCAN7 Register (Offset = 4Ch) [reset = 0h]

TTCAN7 is shown in Figure 21-39 and described in Table 21-27.

Return to Summary Table.

TTCAN7

Figure 21-39. TTCAN7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-27. TTCAN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.21 TTCAN8 Register (Offset = 50h) [reset = 0h]

TTCAN8 is shown in Figure 21-40 and described in Table 21-28.

Return to Summary Table.

TTCAN8

Figure 21-40. TTCAN8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-28. TTCAN8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.22 TTCAN9 Register (Offset = 54h) [reset = 0h]

TTCAN9 is shown in Figure 21-41 and described in Table 21-29.

Return to Summary Table.

TTCAN9

Figure 21-41. TTCAN9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-29. TTCAN9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.23 TTCAN10 Register (Offset = 58h) [reset = 0h]

TTCAN10 is shown in Figure 21-42 and described in Table 21-30.

Return to Summary Table.

TTCAN10

Figure 21-42. TTCAN10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-30. TTCAN10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.24 TTCAN11 Register (Offset = 5Ch) [reset = 0h]

TTCAN11 is shown in Figure 21-43 and described in Table 21-31.

Return to Summary Table.

TTCAN11

Figure 21-43. TTCAN11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-31. TTCAN11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.25 TTCAN12 Register (Offset = 60h) [reset = 0h]

TTCAN12 is shown in Figure 21-44 and described in Table 21-32.

Return to Summary Table.

TTCAN12

Figure 21-44. TTCAN12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-32. TTCAN12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.26 TTCAN13 Register (Offset = 64h) [reset = 0h]

TTCAN13 is shown in Figure 21-45 and described in Table 21-33.

Return to Summary Table.

TTCAN13

Figure 21-45. TTCAN13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-33. TTCAN13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.27 TTCAN14 Register (Offset = 68h) [reset = 0h]

TTCAN14 is shown in Figure 21-46 and described in Table 21-34.

Return to Summary Table.

TTCAN14

Figure 21-46. TTCAN14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-34. TTCAN14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.28 TTCAN15 Register (Offset = 6Ch) [reset = 0h]

TTCAN15 is shown in Figure 21-47 and described in Table 21-35.

Return to Summary Table.

TTCAN15

Figure 21-47. TTCAN15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-35. TTCAN15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.29 TTCAN16 Register (Offset = 70h) [reset = 0h]

TTCAN16 is shown in Figure 21-48 and described in Table 21-36.

Return to Summary Table.

TTCAN16

Figure 21-48. TTCAN16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-36. TTCAN16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.30 TTCAN17 Register (Offset = 74h) [reset = 0h]

TTCAN17 is shown in Figure 21-49 and described in Table 21-37.

Return to Summary Table.

TTCAN17

Figure 21-49. TTCAN17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-37. TTCAN17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.31 TTCAN18 Register (Offset = 78h) [reset = 0h]

TTCAN18 is shown in Figure 21-50 and described in Table 21-38.

Return to Summary Table.

TTCAN18

Figure 21-50. TTCAN18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-38. TTCAN18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.32 TTCAN19 Register (Offset = 7Ch) [reset = 0h]

TTCAN19 is shown in Figure 21-51 and described in Table 21-39.

Return to Summary Table.

TTCAN19

Figure 21-51. TTCAN19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W-																															

Table 21-39. TTCAN19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0	

21.17.33 ABOTR Register (Offset = 80h) [reset = 0h]

ABOTR is shown in Figure 21-52 and described in Table 21-40.

Return to Summary Table.

Auto Bus On Time Register On write access to the CAN control register while Auto-Bus-On timer is running, the Auto-Bus-On procedure will be aborted. During Debug/Suspend mode, running Auto-Bus-On timer will be paused.

Figure 21-52. ABOTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABO_Time																															
R/W-0h																															

Table 21-40. ABOTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ABO_Time	R/W	0h	Number of OCP clock cycles before a Bus-Off recovery sequence is started by clearing the Init bit. This function has to be enabled by setting bit ABO in CAN control register. The Auto-Bus-On timer is realized by a 32 bit counter that starts to count down to zero when the module goes Bus-Off. The counter will be reloaded with the preload value of the ABO time register after this phase.

21.17.34 TXRQ_X Register (Offset = 84h) [reset = 0h]

TXRQ_X is shown in Figure 21-53 and described in Table 21-41.

Return to Summary Table.

Transmission Request X Register Example 1. Bit 0 of the transmission request X register represents byte 0 of the transmission request 1 register. If one or more bits in this byte are set, bit 0 of the transmission request X register will be set.

Figure 21-53. TXRQ_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
TxRqstReg8		TxRqstReg7		TxRqstReg6		TxRqstReg5	
R-0h		R-0h		R-0h		R-0h	
7	6	5	4	3	2	1	0
TxRqstReg4		TxRqstReg3		TxRqstReg2		TxRqstReg1	
R-0h		R-0h		R-0h		R-0h	

Table 21-41. TXRQ_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	TxRqstReg8	R	0h	TxRqstReg8
13-12	TxRqstReg7	R	0h	TxRqstReg7
11-10	TxRqstReg6	R	0h	TxRqstReg6
9-8	TxRqstReg5	R	0h	TxRqstReg5
7-6	TxRqstReg4	R	0h	TxRqstReg4
5-4	TxRqstReg3	R	0h	TxRqstReg3
3-2	TxRqstReg2	R	0h	TxRqstReg2
1-0	TxRqstReg1	R	0h	TxRqstReg1

21.17.35 TXRQ12 Register (Offset = 88h) [reset = 0h]

TXRQ12 is shown in Figure 21-54 and described in Table 21-42.

Return to Summary Table.

Transmission Request Register The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 21-54. TXRQ12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxRqs_1																TxRqs_0															
R-0h																R-0h															

Table 21-42. TXRQ12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TxRqs_1	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.
15-0	TxRqs_0	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

21.17.36 TXRQ34 Register (Offset = 8Ch) [reset = 0h]

TXRQ34 is shown in Figure 21-55 and described in Table 21-43.

Return to Summary Table.

Transmission Request Register The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 21-55. TXRQ34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxRqs_1																TxRqs_0															
R-0h																R-0h															

Table 21-43. TXRQ34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TxRqs_1	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.
15-0	TxRqs_0	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

21.17.37 TXRQ56 Register (Offset = 90h) [reset = 0h]

TXRQ56 is shown in Figure 21-56 and described in Table 21-44.

Return to Summary Table.

Transmission Request Register The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 21-56. TXRQ56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxRqs_1																TxRqs_0															
R-0h																R-0h															

Table 21-44. TXRQ56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TxRqs_1	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.
15-0	TxRqs_0	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

21.17.38 TXRQ78 Register (Offset = 94h) [reset = 0h]

TXRQ78 is shown in Figure 21-57 and described in Table 21-45.

Return to Summary Table.

Transmission Request Register The TXRQ12 to TXRQ78 registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

Figure 21-57. TXRQ78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxRqs_1																TxRqs_0															
R-0h																R-0h															

Table 21-45. TXRQ78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TxRqs_1	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.
15-0	TxRqs_0	R	0h	Transmission request bits (for all message objects) 0h (R/W) = No transmission has been requested for this message object. 1h (R/W) = The transmission of this message object is requested and is not yet done.

21.17.39 NWDAT_X Register (Offset = 98h) [reset = 0h]

NWDAT_X is shown in Figure 21-58 and described in Table 21-46.

Return to Summary Table.

New Data X Register With the new data X register, the CPU can detect if one or more bits in the different new data registers are set. Each register bit represents a group of eight message objects. If at least on of the NewDat bits of these message objects are set, the corresponding bit in the new data X register will be set. Example 1. Bit 0 of the new data X register represents byte 0 of the new data 1 register. If one or more bits in this byte are set, bit 0 of the new data X register will be set.

Figure 21-58. NWDAT_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
NewDatReg8		NewDatReg7		NewDatReg6		NewDatReg5	
R-0h		R-0h		R-0h		R-0h	
7	6	5	4	3	2	1	0
NewDatReg4		NewDatReg3		NewDatReg2		NewDatReg1	
R-0h		R-0h		R-0h		R-0h	

Table 21-46. NWDAT_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	NewDatReg8	R	0h	NewDatReg8
13-12	NewDatReg7	R	0h	NewDatReg7
11-10	NewDatReg6	R	0h	NewDatReg6
9-8	NewDatReg5	R	0h	NewDatReg5
7-6	NewDatReg4	R	0h	NewDatReg4
5-4	NewDatReg3	R	0h	NewDatReg3
3-2	NewDatReg2	R	0h	NewDatReg2
1-0	NewDatReg1	R	0h	NewDatReg1

21.17.40 NWDAT12 Register (Offset = 9Ch) [reset = 0h]

NWDAT12 is shown in Figure 21-59 and described in Table 21-47.

Return to Summary Table.

New Data Register These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 21-59. NWDAT12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewDat_1																NewDat_0															
R-0h																R-0h															

Table 21-47. NWDAT12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NewDat_1	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
15-0	NewDat_0	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

21.17.41 NWDAT34 Register (Offset = A0h) [reset = 0h]

NWDAT34 is shown in Figure 21-60 and described in Table 21-48.

Return to Summary Table.

New Data Register These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 21-60. NWDAT34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewDat_1																NewDat_0															
R-0h																R-0h															

Table 21-48. NWDAT34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NewDat_1	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
15-0	NewDat_0	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

21.17.42 NWDAT56 Register (Offset = A4h) [reset = 0h]

NWDAT56 is shown in Figure 21-61 and described in Table 21-49.

Return to Summary Table.

New Data Register These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 21-61. NWDAT56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewDat_1																NewDat_0															
R-0h																R-0h															

Table 21-49. NWDAT56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NewDat_1	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
15-0	NewDat_0	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

21.17.43 NWDAT78 Register (Offset = A8h) [reset = 0h]

NWDAT78 is shown in Figure 21-62 and described in Table 21-50.

Return to Summary Table.

New Data Register These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

Figure 21-62. NWDAT78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewDat_1																NewDat_0															
R-0h																R-0h															

Table 21-50. NWDAT78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NewDat_1	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
15-0	NewDat_0	R	0h	New Data Bits (for all message objects) 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.

21.17.44 INTPND_X Register (Offset = ACh) [reset = 0h]

INTPND_X is shown in Figure 21-63 and described in Table 21-51.

Return to Summary Table.

Interrupt Pending X Register With the interrupt pending X register, the CPU can detect if one or more bits in the different interrupt pending registers are set. Each bit of this register represents a group of eight message objects. If at least one of the IntPnd bits of these message objects are set, the corresponding bit in the interrupt pending X register will be set. Example 2. Bit 0 of the interrupt pending X register represents byte 0 of the interrupt pending 1 register. If one or more bits in this byte are set, bit 0 of the interrupt pending X register will be set.

Figure 21-63. INTPND_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
IntPndReg8		IntPndReg7		IntPndReg6		IntPndReg5	
R-0h		R-0h		R-0h		R-0h	
7	6	5	4	3	2	1	0
IntPndReg4		IntPndReg3		IntPndReg2		IntPndReg1	
R-0h		R-0h		R-0h		R-0h	

Table 21-51. INTPND_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	IntPndReg8	R	0h	IntPndReg8
13-12	IntPndReg7	R	0h	IntPndReg7
11-10	IntPndReg6	R	0h	IntPndReg6
9-8	IntPndReg5	R	0h	IntPndReg5
7-6	IntPndReg4	R	0h	IntPndReg4
5-4	IntPndReg3	R	0h	IntPndReg3
3-2	IntPndReg2	R	0h	IntPndReg2
1-0	IntPndReg1	R	0h	IntPndReg1

21.17.45 INTPND12 Register (Offset = B0h) [reset = 0h]

INTPND12 is shown in Figure 21-64 and described in Table 21-52.

Return to Summary Table.

Interrupt Pending Register These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-64. INTPND12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntPnd_1																IntPnd_0															
R-0h																R-0h															

Table 21-52. INTPND12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntPnd_1	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.
15-0	IntPnd_0	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

21.17.46 INTPND34 Register (Offset = B4h) [reset = 0h]

INTPND34 is shown in Figure 21-65 and described in Table 21-53.

Return to Summary Table.

Interrupt Pending Register These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-65. INTPND34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntPnd_1																IntPnd_0															
R-0h																R-0h															

Table 21-53. INTPND34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntPnd_1	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.
15-0	IntPnd_0	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

21.17.47 INTPND56 Register (Offset = B8h) [reset = 0h]

INTPND56 is shown in Figure 21-66 and described in Table 21-54.

Return to Summary Table.

Interrupt Pending Register These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-66. INTPND56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntPnd_1																IntPnd_0															
R-0h																R-0h															

Table 21-54. INTPND56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntPnd_1	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.
15-0	IntPnd_0	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

21.17.48 INTPND78 Register (Offset = BCh) [reset = 0h]

INTPND78 is shown in Figure 21-67 and described in Table 21-55.

Return to Summary Table.

Interrupt Pending Register These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-67. INTPND78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntPnd_1																IntPnd_0															
R-0h																R-0h															

Table 21-55. INTPND78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntPnd_1	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.
15-0	IntPnd_0	R	0h	Interrupt Pending Bits (for all message objects) 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt.

21.17.49 MSGVAL_X Register (Offset = C0h) [reset = 0h]

MSGVAL_X is shown in Figure 21-68 and described in Table 21-56.

Return to Summary Table.

Message Valid X Register With the message valid X register, the CPU can detect if one or more bits in the different message valid registers are set. Each bit of this register represents a group of eight message objects. If at least one of the MsgVal bits of these message objects are set, the corresponding bit in the message valid X register will be set. Example 3. Bit 0 of the message valid X register represents byte 0 of the message valid 1 register. If one or more bits in this byte are set, bit 0 of the message valid X register will be set.

Figure 21-68. MSGVAL_X Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
MsgValReg8		MsgValReg7		MsgValReg6		MsgValReg5	
R-0h		R-0h		R-0h		R-0h	
7	6	5	4	3	2	1	0
MsgValReg4		MsgValReg3		MsgValReg2		MsgValReg1	
R-0h		R-0h		R-0h		R-0h	

Table 21-56. MSGVAL_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-14	MsgValReg8	R	0h	MsgValReg8
13-12	MsgValReg7	R	0h	MsgValReg7
11-10	MsgValReg6	R	0h	MsgValReg6
9-8	MsgValReg5	R	0h	MsgValReg5
7-6	MsgValReg4	R	0h	MsgValReg4
5-4	MsgValReg3	R	0h	MsgValReg3
3-2	MsgValReg2	R	0h	MsgValReg2
1-0	MsgValReg1	R	0h	MsgValReg1

21.17.50 MSGVAL12 Register (Offset = C4h) [reset = 0h]

MSGVAL12 is shown in Figure 21-69 and described in Table 21-57.

Return to Summary Table.

Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-69. MSGVAL12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MsgVal_1																MsgVal_0															
R-0h																R-0h															

Table 21-57. MSGVAL12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MsgVal_1	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.
15-0	MsgVal_0	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

21.17.51 MSGVAL34 Register (Offset = C8h) [reset = 0h]

MSGVAL34 is shown in Figure 21-70 and described in Table 21-58.

Return to Summary Table.

Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-70. MSGVAL34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MsgVal_1																MsgVal_0															
R-0h																R-0h															

Table 21-58. MSGVAL34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MsgVal_1	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.
15-0	MsgVal_0	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

21.17.52 MSGVAL56 Register (Offset = CCh) [reset = 0h]

MSGVAL56 is shown in Figure 21-71 and described in Table 21-59.

Return to Summary Table.

Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-71. MSGVAL56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MsgVal_1																MsgVal_0															
R-0h																R-0h															

Table 21-59. MSGVAL56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MsgVal_1	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.
15-0	MsgVal_0	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

21.17.53 MSGVAL78 Register (Offset = D0h) [reset = 0h]

MSGVAL78 is shown in Figure 21-72 and described in Table 21-60.

Return to Summary Table.

Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

Figure 21-72. MSGVAL78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MsgVal_1																MsgVal_0															
R-0h																R-0h															

Table 21-60. MSGVAL78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MsgVal_1	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.
15-0	MsgVal_0	R	0h	Message valid bits (for all message objects) 0h (R/W) = This message object is ignored by the message handler. 1h (R/W) = This message object is configured and will be considered by the message handler.

21.17.54 Reserved_2 Register (Offset = D4h) [reset = 0h]

Reserved_2 is shown in Figure 21-73 and described in Table 21-61.

Return to Summary Table.

Figure 21-73. Reserved_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewBitfield																															
R-																															

Table 21-61. Reserved_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NewBitfield	R	0	

21.17.55 INTMUX12 Register (Offset = D8h) [reset = 0h]

INTMUX12 is shown in Figure 21-74 and described in Table 21-62.

Return to Summary Table.

Interrupt Pending Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 21-74. INTMUX12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntMux_1																IntMux_0															
R-0h																R-0h															

Table 21-62. INTMUX12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntMux_1	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.
15-0	IntMux_0	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

21.17.56 INTMUX34 Register (Offset = DCh) [reset = 0h]

INTMUX34 is shown in Figure 21-75 and described in Table 21-63.

Return to Summary Table.

Interrupt Pending Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 21-75. INTMUX34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntMux_1																IntMux_0															
R-0h																R-0h															

Table 21-63. INTMUX34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntMux_1	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.
15-0	IntMux_0	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

21.17.57 INTMUX56 Register (Offset = E0h) [reset = 0h]

INTMUX56 is shown in Figure 21-76 and described in Table 21-64.

Return to Summary Table.

Interrupt Pending Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 21-76. INTMUX56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntMux_1																IntMux_0															
R-0h																R-0h															

Table 21-64. INTMUX56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntMux_1	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.
15-0	IntMux_0	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

21.17.58 INTMUX78 Register (Offset = E4h) [reset = 0h]

INTMUX78 is shown in Figure 21-77 and described in Table 21-65.

Return to Summary Table.

Interrupt Pending Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

Figure 21-77. INTMUX78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntMux_1																IntMux_0															
R-0h																R-0h															

Table 21-65. INTMUX78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IntMux_1	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.
15-0	IntMux_0	R	0h	Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) 0h (R/W) = DCAN0INT line is active if corresponding IntPnd flag is one. 1h (R/W) = DCAN1INT line is active if corresponding IntPnd flag is one.

21.17.59 Reserved_3 Register (Offset = E8h) [reset = 0h]

Reserved_3 is shown in Figure 21-78 and described in Table 21-66.

Return to Summary Table.

Figure 21-78. Reserved_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-66. Reserved_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.60 Reserved_4 Register (Offset = ECh) [reset = 0h]

Reserved_4 is shown in Figure 21-79 and described in Table 21-67.

Return to Summary Table.

Figure 21-79. Reserved_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-67. Reserved_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.61 Reserved_5 Register (Offset = F0h) [reset = 0h]

Reserved_5 is shown in Figure 21-80 and described in Table 21-68.

Return to Summary Table.

Figure 21-80. Reserved_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-68. Reserved_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.62 Reserved_6 Register (Offset = F4h) [reset = 0h]

Reserved_6 is shown in Figure 21-81 and described in Table 21-69.

Return to Summary Table.

Figure 21-81. Reserved_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-69. Reserved_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.63 Reserved_7 Register (Offset = F8h) [reset = 0h]

Reserved_7 is shown in Figure 21-82 and described in Table 21-70.

Return to Summary Table.

Figure 21-82. Reserved_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-70. Reserved_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.64 Reserved_8 Register (Offset = FCh) [reset = 0h]

Reserved_8 is shown in Figure 21-83 and described in Table 21-71.

Return to Summary Table.

Figure 21-83. Reserved_8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-71. Reserved_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.65 IF1CMD Register (Offset = 100h) [reset = 1h]

IF1CMD is shown in Figure 21-84 and described in Table 21-72.

Return to Summary Table.

IF1 Command Register The IF1 Command Register (IF1CMD) configures and initiates the transfer between the IF1 register sets and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the CPU writes the message number to bits [7:0] of the IF1 command register. With this write operation, the Busy bit is automatically set to '1' to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the Busy bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the CPU writes to both IF1 command registers consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While Busy bit is one, IF1 register sets are write protected. For debug support, the auto clear functionality of the IF1 command registers (clear of DMAactive flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] of the IF1 command register, the message handler may access an implemented (valid) message object instead.

Figure 21-84. IF1CMD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
WR_RD	Mask	Arb	Control	ClrIntPnd	TxRqst_NewDat	Data_A	Data_B
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h
15	14	13	12	11	10	9	8
Busy	DMAactive	RESERVED					
R/WP-0h	R/WP-0h	R-0h					
7	6	5	4	3	2	1	0
Message_Number							
R/WP-1h							

Table 21-72. IF1CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	WR_RD	R/WP	0h	Write/Read 0h (R/W) = Direction = Read: Transfer direction is from the message object addressed by Message Number (Bits [7:0]) to the IF1 register set. 1h (R/W) = Direction = Write: Transfer direction is from the IF1 register set to the message object addressed by Message Number (Bits [7:0]).
22	Mask	R/WP	0h	Access mask bits 0h (R/W) = Mask bits will not be changed 1h (R/W) = Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the IF1 register set. Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF1 register set to the message object addressed by Message Number (Bits [7:0]).

Table 21-72. IF1CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	Arb	R/WP	0h	Access arbitration bits 0h (R/W) = Arbitration bits will not be changed 1h (R/W) = Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1 register set. Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF1 register set to the message object addressed by Message Number (Bits [7:0]).
20	Control	R/WP	0h	Access control bits. If the TxRqst/NewDat bit in this register(Bit [18]) is set, the TxRqst/ NewDat bit in the IF1 message control register will be ignored. 0h (R/W) = Control bits will not be changed 1h (R/W) = Direction = Read: The message control bits will be transferred from the message object addressed by message number (Bits [7:0]) to the IF1 register set. Direction = Write: The message control bits will be transferred from the IF1 register set to the message object addressed by message number (Bits [7:0]).
19	ClrIntPnd	R/WP	0h	Clear interrupt pending bit 0h (R/W) = IntPnd bit will not be changed 1h (R/W) = Direction = Read: Clears IntPnd bit in the message object. Direction = Write: This bit is ignored. Copying of IntPnd flag from IF1 Registers to message RAM can only be controlled by the control flag (Bit [20]).
18	TxRqst_NewDat	R/WP	0h	Access transmission request bit. Note: If a CAN transmission is requested by setting TxRqst/NewDat in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in IF1 message control Register. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the IF1 message control register always reflect the status before resetting them. 0h (R/W) = Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the control bit. 1h (R/W) = Direction = Read: Clears NewDat bit in the message object. Direction = Write: Sets TxRqst/NewDat in message object.
17	Data_A	R/WP	0h	Access Data Bytes 0 to 3. 0h (R/W) = Data Bytes 0-3 will not be changed. 1h (R/W) = Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the Message Number (Bits [7:0]) to the corresponding IF1 register set. Direction = Write: The data bytes 0-3 will be transferred from the IF1 register set to the message object addressed by the Message Number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
16	Data_B	R/WP	0h	Access Data Bytes 4 to 7. 0h (R/W) = Data Bytes 4-7 will not be changed. 1h (R/W) = Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1 register set. Direction = Write: The data bytes 4-7 will be transferred from the IF1 register set to the message object addressed by message number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
15	Busy	R/WP	0h	Busy flag. This bit is set to one after the message number has been written to bits 7 to 0. IF1 register set will be write protected. The bit is cleared after read/write action has been finished. 0h (R/W) = No transfer between IF1 register set and message RAM is in progress. 1h (R/W) = Transfer between IF1 register set and message RAM is in progress.

Table 21-72. IF1CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	DMAactive	R/WP	0h	Activation of DMA feature for subsequent internal IF1 update. Note: Due to the auto reset feature of the DMAactive bit, this bit has to be set for each subsequent DMA cycle separately. 0h (R/W) = DMA request line is independent of IF1 activities. 1h (R/W) = DMA is requested after completed transfer between IF1 register set and message RAM. The DMA request remains active until the first read or write to one of the IF1 registers; an exception is a write to Message Number (Bits [7:0]) when DMAactive is one.
13-8	RESERVED	R	0h	
7-0	Message_Number	R/WP	1h	Number of message object in message RAM which is used for data transfer. 0h (R/W) = Invalid message numbers. 1h (R/W) = Valid message numbers (value 01 to 80). 50h (R/W) = Valid message number. 51h (R/W) = Invalid message numbers (value 81 to FF).

21.17.66 IF1MSK Register (Offset = 104h) [reset = FFFFFFFFh]

IF1MSK is shown in Figure 21-85 and described in Table 21-73.

Return to Summary Table.

IF1 Mask Register The bits of the IF1 mask registers mirror the mask bits of a message object. While Busy bit of IF1 command register is one, IF1 register set is write protected.

Figure 21-85. IF1MSK Register

31	30	29	28	27	26	25	24
MXtd	MDir	RESERVED	Msk				
R/WP-1h	R/WP-1h	R-1h	R/WP-1FFFFFFFh				
23	22	21	20	19	18	17	16
Msk							
R/WP-1FFFFFFFh							
15	14	13	12	11	10	9	8
Msk							
R/WP-1FFFFFFFh							
7	6	5	4	3	2	1	0
Msk							
R/WP-1FFFFFFFh							

Table 21-73. IF1MSK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MXtd	R/WP	1h	Mask Extended Identifier. When 11 bit (standard) identifiers are used for a message object, the identifiers of received data frames are written into bits ID28 to ID18. For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 are considered. 0h (R/W) = The extended identifier bit (IDE) has no effect on the acceptance filtering. 1h (R/W) = The extended identifier bit (IDE) is used for acceptance filtering.
30	MDir	R/WP	1h	Mask Message Direction 0h (R/W) = The message direction bit (Dir) has no effect on the acceptance filtering. 1h (R/W) = The message direction bit (Dir) is used for acceptance filtering.
29	RESERVED	R	1h	
28-0	Msk	R/WP	1FFFFFFFh	Identifier Mask 0h (R/W) = The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1h (R/W) = The corresponding bit in the identifier of the message object is used for acceptance filtering.

21.17.67 IF1ARB Register (Offset = 108h) [reset = 0h]

IF1ARB is shown in Figure 21-86 and described in Table 21-74.

Return to Summary Table.

IF1 Arbitration Register The bits of the IF1 arbitration registers mirror the arbitration bits of a message object. While Busy bit of IF1 command register is one, IF1 register set is write protected.

Figure 21-86. IF1ARB Register

31	30	29	28	27	26	25	24
MsgVal	Xtd	Dir	ID28_to_ID0				
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h				
23	22	21	20	19	18	17	16
ID28_to_ID0							
R/WP-0h							
15	14	13	12	11	10	9	8
ID28_to_ID0							
R/WP-0h							
7	6	5	4	3	2	1	0
ID28_to_ID0							
R/WP-0h							

Table 21-74. IF1ARB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MsgVal	R/WP	0h	Message valid. The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID28 to ID0, the control bits Xtd, Dir or DLC3 to DLC0 are modified, or if the messages object is no longer required. 0h (R/W) = The message object is ignored by the message handler. 1h (R/W) = The message object is to be used by the message handler.
30	Xtd	R/WP	0h	Extended identifier 0h (R/W) = The 11-bit (standard) Identifier is used for this message object. 1h (R/W) = The 29-bit (extended) Identifier is used for this message object.
29	Dir	R/WP	0h	Message direction 0h (R/W) = Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1h (R/W) = Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
28-0	ID28_to_ID0	R/WP	0h	Message identifier. ID28 to ID0 is equal to 29 bit identifier (extended frame) ID28 to ID18 is equal to 11 bit identifier (standard frame)

21.17.68 IF1MCTL Register (Offset = 10Ch) [reset = 0h]

IF1MCTL is shown in Figure 21-87 and described in Table 21-75.

Return to Summary Table.

IF1 Message Control Register The bits of the IF1 message control registers mirror the message control bits of a message object. While Busy bit of IF1 command register is one, IF1 register set is write protected.

Figure 21-87. IF1MCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R/WP-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/WP-0h							
15	14	13	12	11	10	9	8
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h
7	6	5	4	3	2	1	0
EoB	RESERVED			DLC			
R/WP-0h	R/WP-0h			R/WP-0h			

Table 21-75. IF1MCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/WP	0h	
15	NewDat	R/WP	0h	New data 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
14	MsgLst	R/WP	0h	Message lost (only valid for message objects with direction = receive) 0h (R/W) = No message lost since the last time when this bit was reset by the CPU. 1h (R/W) = The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	IntPnd	R/WP	0h	Interrupt pending 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMask	R/WP	0h	Use acceptance mask. If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one. 0h (R/W) = Mask ignored 1h (R/W) = Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering
11	TxIE	R/WP	0h	Transmit interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful transmission of a frame. 1h (R/W) = IntPnd will be triggered after the successful transmission of a frame.

Table 21-75. IF1MCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RxIE	R/WP	0h	Receive interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful reception of a frame. 1h (R/W) = IntPnd will be triggered after the successful reception of a frame.
9	RmtEn	R/WP	0h	Remote enable 0h (R/W) = At the reception of a remote frame, TxRqst is not changed. 1h (R/W) = At the reception of a remote frame, TxRqst is set.
8	TxRqst	R/WP	0h	Transmit request 0h (R/W) = This message object is not waiting for a transmission. 1h (R/W) = The transmission of this message object is requested and is not yet done.
7	EoB	R/WP	0h	Data frame has 0 to 8 data bits. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. 0h (R/W) = Data frame has 8 data bytes. 1h (R/W) = Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
6-4	RESERVED	R/WP	0h	
3-0	DLC	R/WP	0h	Data length code. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. 8h (R/W) = Data frame has 0 8 data bits. 15h (R/W) = Data frame has 8 data bytes.

21.17.69 IF1DATA Register (Offset = 110h) [reset = 0h]

IF1DATA is shown in Figure 21-88 and described in Table 21-76.

Return to Summary Table.

IF1 Data A Register The data bytes of CAN messages are stored in the IF1 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 21-88. IF1DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data_3								Data_2								Data_1								Data_0							
R/WP-0h								R/WP-0h								R/WP-0h								R/WP-0h							

Table 21-76. IF1DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Data_3	R/WP	0h	Data 3.
23-16	Data_2	R/WP	0h	Data 2.
15-8	Data_1	R/WP	0h	Data 1.
7-0	Data_0	R/WP	0h	Data 0.

21.17.70 IF1DATB Register (Offset = 114h) [reset = 0h]

IF1DATB is shown in Figure 21-89 and described in Table 21-77.

Return to Summary Table.

IF1 Data B Register The data bytes of CAN messages are stored in the IF1 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 21-89. IF1DATB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data_7								Data_6								Data_5								Data_4							
R/WP-0h								R/WP-0h								R/WP-0h								R/WP-0h							

Table 21-77. IF1DATB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Data_7	R/WP	0h	Data 7.
23-16	Data_6	R/WP	0h	Data 6.
15-8	Data_5	R/WP	0h	Data 5.
7-0	Data_4	R/WP	0h	Data 4.

21.17.71 Reserved_9 Register (Offset = 118h) [reset = 0h]

Reserved_9 is shown in Figure 21-90 and described in Table 21-78.

Return to Summary Table.

Figure 21-90. Reserved_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-78. Reserved_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.72 Reserved_10 Register (Offset = 11Ch) [reset = 0h]

Reserved_10 is shown in Figure 21-91 and described in Table 21-79.

Return to Summary Table.

Figure 21-91. Reserved_10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-79. Reserved_10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.73 IF2CMD Register (Offset = 120h) [reset = 1h]

IF2CMD is shown in Figure 21-92 and described in Table 21-80.

Return to Summary Table.

IF2 Command Register The IF2 Command Register (IF1CMD) configures and initiates the transfer between the IF2 register sets and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the CPU writes the message number to bits [7:0] of the IF2 command register. With this write operation, the Busy bit is automatically set to '1' to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the Busy bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the CPU writes to both IF2 command registers consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While Busy bit is one, IF2 register sets are write protected. For debug support, the auto clear functionality of the IF2 command registers (clear of DMAactive flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] of the IF2 command register, the message handler may access an implemented (valid) message object instead.

Figure 21-92. IF2CMD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
WR_RD	Mask	Arb	Control	ClrIntPnd	TxRqst_NewDat	Data_A	Data_B
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h
15	14	13	12	11	10	9	8
Busy	DMAactive	RESERVED					
R/WP-0h	R/WP-0h	R-0h					
7	6	5	4	3	2	1	0
Message_Number							
R/WP-1h							

Table 21-80. IF2CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	WR_RD	R/WP	0h	Write/Read 0h (R/W) = Direction = Read: Transfer direction is from the message object addressed by Message Number (Bits [7:0]) to the IF2 register set. 1h (R/W) = Direction = Write: Transfer direction is from the IF2 register set to the message object addressed by Message Number (Bits [7:0]).
22	Mask	R/WP	0h	Access mask bits 0h (R/W) = Mask bits will not be changed 1h (R/W) = Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the IF2 register set. Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF2 register set to the message object addressed by Message Number (Bits [7:0]).

Table 21-80. IF2CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	Arb	R/WP	0h	Access arbitration bits 0h (R/W) = Arbitration bits will not be changed 1h (R/W) = Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF2 register set. Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF2 register set to the message object addressed by Message Number (Bits [7:0]).
20	Control	R/WP	0h	Access control bits. If the TxRqst/NewDat bit in this register(Bit [18]) is set, the TxRqst/ NewDat bit in the IF2 message control register will be ignored. 0h (R/W) = Control bits will not be changed 1h (R/W) = Direction = Read: The message control bits will be transferred from the message object addressed by message number (Bits [7:0]) to the IF2 register set. Direction = Write: The message control bits will be transferred from the IF2 register set to the message object addressed by message number (Bits [7:0]).
19	ClrIntPnd	R/WP	0h	Clear interrupt pending bit 0h (R/W) = IntPnd bit will not be changed 1h (R/W) = Direction = Read: Clears IntPnd bit in the message object. Direction = Write: This bit is ignored. Copying of IntPnd flag from IF2 Registers to message RAM can only be controlled by the control flag (Bit [20]).
18	TxRqst_NewDat	R/WP	0h	Access transmission request bit. Note: If a CAN transmission is requested by setting TxRqst/NewDat in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in IF2 message control Register. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the IF2 message control register always reflect the status before resetting them. 0h (R/W) = Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the control bit. 1h (R/W) = Direction = Read: Clears NewDat bit in the message object. Direction = Write: Sets TxRqst/NewDat in message object.
17	Data_A	R/WP	0h	Access Data Bytes 0 to 3. 0h (R/W) = Data Bytes 0-3 will not be changed. 1h (R/W) = Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the Message Number (Bits [7:0]) to the corresponding IF2 register set. Direction = Write: The data bytes 0-3 will be transferred from the IF2 register set to the message object addressed by the Message Number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
16	Data_B	R/WP	0h	Access Data Bytes 4 to 7. 0h (R/W) = Data Bytes 4-7 will not be changed. 1h (R/W) = Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF2 register set. Direction = Write: The data bytes 4-7 will be transferred from the IF2 register set to the message object addressed by message number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
15	Busy	R/WP	0h	Busy flag. This bit is set to one after the message number has been written to bits 7 to 0. IF2 register set will be write protected. The bit is cleared after read/write action has been finished. 0h (R/W) = No transfer between IF2 register set and message RAM is in progress. 1h (R/W) = Transfer between IF2 register set and message RAM is in progress.

Table 21-80. IF2CMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	DMAactive	R/WP	0h	Activation of DMA feature for subsequent internal IF2 update. Note: Due to the auto reset feature of the DMAactive bit, this bit has to be set for each subsequent DMA cycle separately. 0h (R/W) = DMA request line is independent of IF2 activities. 1h (R/W) = DMA is requested after completed transfer between IF2 register set and message RAM. The DMA request remains active until the first read or write to one of the IF2 registers; an exception is a write to Message Number (Bits [7:0]) when DMAactive is one.
13-8	RESERVED	R	0h	
7-0	Message_Number	R/WP	1h	Number of message object in message RAM which is used for data transfer. 0h (R/W) = Invalid message number. 1h (R/W) = Valid message numbers (values 01 to 80). 50h (R/W) = Valid message number. 51h (R/W) = Invalid message numbers (values 81 to FF).

21.17.74 IF2MSK Register (Offset = 124h) [reset = FFFFFFFFh]

IF2MSK is shown in Figure 21-93 and described in Table 21-81.

Return to Summary Table.

IF2 Mask Register The bits of the IF2 mask registers mirror the mask bits of a message object. While Busy bit of IF2 command register is one, IF2 register set is write protected.

Figure 21-93. IF2MSK Register

31	30	29	28	27	26	25	24
MXtd	MDir	RESERVED	Msk				
R/WP-1h	R/WP-1h	R-1h	R/WP-1FFFFFFFh				
23	22	21	20	19	18	17	16
Msk							
R/WP-1FFFFFFFh							
15	14	13	12	11	10	9	8
Msk							
R/WP-1FFFFFFFh							
7	6	5	4	3	2	1	0
Msk							
R/WP-1FFFFFFFh							

Table 21-81. IF2MSK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MXtd	R/WP	1h	Mask Extended Identifier. When 11 bit (standard) identifiers are used for a message object, the identifiers of received data frames are written into bits ID28 to ID18. For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 are considered. 0h (R/W) = The extended identifier bit (IDE) has no effect on the acceptance filtering. 1h (R/W) = The extended identifier bit (IDE) is used for acceptance filtering.
30	MDir	R/WP	1h	Mask Message Direction 0h (R/W) = The message direction bit (Dir) has no effect on the acceptance filtering. 1h (R/W) = The message direction bit (Dir) is used for acceptance filtering.
29	RESERVED	R	1h	
28-0	Msk	R/WP	1FFFFFFFh	Identifier Mask 0h (R/W) = The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1h (R/W) = The corresponding bit in the identifier of the message object is used for acceptance filtering.

21.17.75 IF2ARB Register (Offset = 128h) [reset = 0h]

IF2ARB is shown in Figure 21-94 and described in Table 21-82.

Return to Summary Table.

IF2 Arbitration Register The bits of the IF2 arbitration registers mirror the arbitration bits of a message object. While Busy bit of IF2 command register is one, IF2 register set is write protected.

Figure 21-94. IF2ARB Register

31	30	29	28	27	26	25	24
MsgVal	Xtd	Dir	ID28_to_ID0				
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h				
23	22	21	20	19	18	17	16
ID28_to_ID0							
R/WP-0h							
15	14	13	12	11	10	9	8
ID28_to_ID0							
R/WP-0h							
7	6	5	4	3	2	1	0
ID28_to_ID0							
R/WP-0h							

Table 21-82. IF2ARB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MsgVal	R/WP	0h	Message valid. The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID28 to ID0, the control bits Xtd, Dir or DLC3 to DLC0 are modified, or if the messages object is no longer required. 0h (R/W) = The message object is ignored by the message handler. 1h (R/W) = The message object is to be used by the message handler.
30	Xtd	R/WP	0h	Extended identifier 0h (R/W) = The 11-bit (standard) Identifier is used for this message object. 1h (R/W) = The 29-bit (extended) Identifier is used for this message object.
29	Dir	R/WP	0h	Message direction 0h (R/W) = Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1h (R/W) = Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
28-0	ID28_to_ID0	R/WP	0h	Message identifier. ID28 to ID0 is equal to 29-bit identifier (extended frame) ID28 to ID18 is equal to 11-bit identifier (standard frame)

21.17.76 IF2MCTL Register (Offset = 12Ch) [reset = 0h]

IF2MCTL is shown in Figure 21-95 and described in Table 21-83.

Return to Summary Table.

IF2 Message Control Register The bits of the IF2 message control registers mirror the message control bits of a message object. While Busy bit of IF2 command register is one, IF2 register set is write protected.

Figure 21-95. IF2MCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R/WP-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/WP-0h							
15	14	13	12	11	10	9	8
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h
7	6	5	4	3	2	1	0
EoB	RESERVED			DLC			
R/WP-0h	R/WP-0h			R/WP-0h			

Table 21-83. IF2MCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/WP	0h	
15	NewDat	R/WP	0h	New data 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
14	MsgLst	R/WP	0h	Message lost (only valid for message objects with direction = receive) 0h (R/W) = No message lost since the last time when this bit was reset by the CPU. 1h (R/W) = The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	IntPnd	R/WP	0h	Interrupt pending 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMask	R/WP	0h	Use acceptance mask. If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one. 0h (R/W) = Mask ignored 1h (R/W) = Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering
11	TxIE	R/WP	0h	Transmit interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful transmission of a frame. 1h (R/W) = IntPnd will be triggered after the successful transmission of a frame.

Table 21-83. IF2MCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RxIE	R/WP	0h	Receive interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful reception of a frame. 1h (R/W) = IntPnd will be triggered after the successful reception of a frame.
9	RmtEn	R/WP	0h	Remote enable 0h (R/W) = At the reception of a remote frame, TxRqst is not changed. 1h (R/W) = At the reception of a remote frame, TxRqst is set.
8	TxRqst	R/WP	0h	Transmit request 0h (R/W) = This message object is not waiting for a transmission. 1h (R/W) = The transmission of this message object is requested and is not yet done.
7	EoB	R/WP	0h	Data frame has 0 to 8 data bits. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. 0h (R/W) = Data frame has 8 data bytes. 1h (R/W) = Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
6-4	RESERVED	R/WP	0h	
3-0	DLC	R/WP	0h	Data length code. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. 8h (R/W) = Data frame has 0 8 data bits. 15h (R/W) = Data frame has 8 data bytes.

21.17.77 IF2DATA Register (Offset = 130h) [reset = 0h]

IF2DATA is shown in Figure 21-96 and described in Table 21-84.

Return to Summary Table.

IF2 Data A Register The data bytes of CAN messages are stored in the IF2 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 21-96. IF2DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data_3								Data_2								Data_1								Data_0							
R/WP-0h								R/WP-0h								R/WP-0h								R/WP-0h							

Table 21-84. IF2DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Data_3	R/WP	0h	Data 3.
23-16	Data_2	R/WP	0h	Data 2.
15-8	Data_1	R/WP	0h	Data 1.
7-0	Data_0	R/WP	0h	Data 0.

21.17.78 IF2DATB Register (Offset = 134h) [reset = 0h]

IF2DATB is shown in Figure 21-97 and described in Table 21-85.

Return to Summary Table.

IF2 Data B Register The data bytes of CAN messages are stored in the IF2 registers in the following order: (1) In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. (2) In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 21-97. IF2DATB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data_7								Data_6								Data_5								Data_4							
R/WP-0h								R/WP-0h								R/WP-0h								R/WP-0h							

Table 21-85. IF2DATB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Data_7	R/WP	0h	Data 7.
23-16	Data_6	R/WP	0h	Data 6.
15-8	Data_5	R/WP	0h	Data 5.
7-0	Data_4	R/WP	0h	Data 4.

21.17.79 Reserved_11 Register (Offset = 138h) [reset = 0h]

Reserved_11 is shown in Figure 21-98 and described in Table 21-86.

Return to Summary Table.

Figure 21-98. Reserved_11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-86. Reserved_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.80 Reserved_12 Register (Offset = 13Ch) [reset = 0h]

Reserved_12 is shown in Figure 21-99 and described in Table 21-87.

Return to Summary Table.

Figure 21-99. Reserved_12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-87. Reserved_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.81 IF3OBS Register (Offset = 140h) [reset = 0h]

IF3OBS is shown in Figure 21-100 and described in Table 21-88.

Return to Summary Table.

IF3 Observation Register The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by CPU. The observation flags (Bits [4:0]) in the IF3 observation register are used to determine, which data sections of the IF3 interface register set have to be read in order to complete a DMA read cycle. After all marked data sections are read, the DCAN is enabled to update the IF3 interface register set with new data. Any access order of single bytes or half-words is supported. When using byte or half-word accesses, a data section is marked as completed, if all bytes are read. Note: If IF3 Update Enable is used and no Observation flag is set, the corresponding message objects will be copied to IF3 without activating the DMA request line and without waiting for DMA read accesses. A write access to this register aborts a pending DMA cycle by resetting the DMA line and enables updating of IF3 interface register set with new data. To avoid data inconsistency, the DMA controller should be disabled before reconfiguring IF3 observation register. The status of the current read-cycle can be observed via status flags (Bits [12:8]). If an interrupt line is available for IF3, an interrupt will be generated by IF3Upd flag. See the device-specific data sheet for the availability of this interrupt source. With this interrupt, the observation status bits and the IF3Upd bit could be used by the application to realize the notification about new IF3 content in polling or interrupt mode.

Figure 21-100. IF3OBS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
IF3_Upd	RESERVED		IF3_SDB	IF3_SDA	IF3_SC	IF3_SA	IF3_SM
R-0h	R-0h		R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED			DataB	DataA	Ctrl	Arb	Mask
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-88. IF3OBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	IF3_Upd	R	0h	IF3 Update Data 0h (R/W) = No new data has been loaded since last IF3 read. 1h (R/W) = New data has been loaded since last IF3 read.
14-13	RESERVED	R	0h	
12	IF3_SDB	R	0h	IF3 Status of Data B read access 0h (R/W) = All Data B bytes are already read out, or are not marked to be read. 1h (R/W) = Data B section has still data to be read out.
11	IF3_SDA	R	0h	IF3 Status of Data A read access 0h (R/W) = All Data A bytes are already read out, or are not marked to be read. 1h (R/W) = Data A section has still data to be read out.
10	IF3_SC	R	0h	IF3 Status of control bits read access 0h (R/W) = All control section bytes are already read out, or are not marked to be read. 1h (R/W) = Control section has still data to be read out.

Table 21-88. IF3OBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	IF3_SA	R	0h	IF3 Status of Arbitration data read access 0h (R/W) = All Arbitration data bytes are already read out, or are not marked to be read. 1h (R/W) = Arbitration section has still data to be read out.
8	IF3_SM	R	0h	IF3 Status of Mask data read access 0h (R/W) = All mask data bytes are already read out, or are not marked to be read. 1h (R/W) = Mask section has still data to be read out.
7-5	RESERVED	R	0h	
4	DataB	R/W	0h	Data B read observation 0h (R/W) = Data B section has not to be read. 1h (R/W) = Data B section has to be read to enable next IF3 update.
3	DataA	R/W	0h	Data A read observation 0h (R/W) = Data A section has not to be read. 1h (R/W) = Data A section has to be read to enable next IF3 update.
2	Ctrl	R/W	0h	Ctrl read observation 0h (R/W) = Ctrl section has not to be read. 1h (R/W) = Ctrl section has to be read to enable next IF3 update.
1	Arb	R/W	0h	Arbitration data read observation 0h (R/W) = Arbitration data has not to be read. 1h (R/W) = Arbitration data has to be read to enable next IF3 update.
0	Mask	R/W	0h	Mask data read observation 0h (R/W) = Mask data has not to be read. 1h (R/W) = Mask data has to be read to enable next IF3 update.

21.17.82 IF3MSK Register (Offset = 144h) [reset = FFFFFFFFh]

IF3MSK is shown in Figure 21-101 and described in Table 21-89.

Return to Summary Table.

IF3 Mask Register

Figure 21-101. IF3MSK Register

31	30	29	28	27	26	25	24
MXtd	MDir	RESERVED	Msk				
R-1h	R-1h	R-1h	R/WP-1FFFFFFFh				
23	22	21	20	19	18	17	16
Msk							
R/WP-1FFFFFFFh							
15	14	13	12	11	10	9	8
Msk							
R/WP-1FFFFFFFh							
7	6	5	4	3	2	1	0
Msk							
R/WP-1FFFFFFFh							

Table 21-89. IF3MSK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MXtd	R	1h	Mask Extended Identifier. When 11 bit (standard) identifiers are used for a message object, the identifiers of received data frames are written into bits ID28 to ID18. For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 are considered. 0h (R/W) = The extended identifier bit (IDE) has no effect on the acceptance filtering. 1h (R/W) = The extended identifier bit (IDE) is used for acceptance filtering.
30	MDir	R	1h	Mask Message Direction 0h (R/W) = The message direction bit (Dir) has no effect on the acceptance filtering. 1h (R/W) = The message direction bit (Dir) is used for acceptance filtering.
29	RESERVED	R	1h	
28-0	Msk	R/WP	1FFFFFFFh	Identifier Mask 0h (R/W) = The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1h (R/W) = The corresponding bit in the identifier of the message object is used for acceptance filtering.

21.17.83 IF3ARB Register (Offset = 148h) [reset = 0h]

IF3ARB is shown in Figure 21-102 and described in Table 21-90.

Return to Summary Table.

IF3 Arbitration Register

Figure 21-102. IF3ARB Register

31	30	29	28	27	26	25	24
MsgVal	Xtd	Dir	ID28_to_ID0				
R-0h	R-0h	R-0h	R-0h				
23	22	21	20	19	18	17	16
ID28_to_ID0							
R-0h							
15	14	13	12	11	10	9	8
ID28_to_ID0							
R-0h							
7	6	5	4	3	2	1	0
ID28_to_ID0							
R-0h							

Table 21-90. IF3ARB Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MsgVal	R	0h	Message Valid. The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID28 to ID0, the control bits Xtd, Dir or DLC3 to DLC0 are modified, or if the messages object is no longer required. 0h (R/W) = The message object is ignored by the message handler. 1h (R/W) = The message object is to be used by the message handler.
30	Xtd	R	0h	Extended Identifier 0h (R/W) = The 11-bit (standard) Identifier is used for this message object. 1h (R/W) = The 29-bit (extended) Identifier is used for this message object.
29	Dir	R	0h	Message Direction 0h (R/W) = Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1h (R/W) = Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
28-0	ID28_to_ID0	R	0h	Message Identifier. ID28 to ID0 is equal to 29 bit Identifier (extended frame). ID28 to ID18 is equal to 11 bit Identifier (standard frame).

21.17.84 IF3MCTL Register (Offset = 14Ch) [reset = 0h]

IF3MCTL is shown in Figure 21-103 and described in Table 21-91.

Return to Summary Table.

IF3 Message Control Register

Figure 21-103. IF3MCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
NewDat	MsgLst	IntPnd	UMask	TxIE	RxIE	RmtEn	TxRqst
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
EoB	RESERVED			DLC			
R-0h	R-0h			R-0h			

Table 21-91. IF3MCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	NewDat	R	0h	New Data 0h (R/W) = No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. 1h (R/W) = The message handler or the CPU has written new data into the data portion of this message object.
14	MsgLst	R	0h	Message Lost (only valid for message objects with direction = receive) 0h (R/W) = No message lost since the last time when this bit was reset by the CPU. 1h (R/W) = The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	IntPnd	R	0h	Interrupt Pending 0h (R/W) = This message object is not the source of an interrupt. 1h (R/W) = This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMask	R	0h	Use Acceptance Mask 0h (R/W) = Mask ignored 1h (R/W) = Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering. If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.
11	TxIE	R	0h	Transmit Interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful transmission of a frame. 1h (R/W) = IntPnd will be triggered after the successful transmission of a frame.

Table 21-91. IF3MCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RxIE	R	0h	Receive Interrupt enable 0h (R/W) = IntPnd will not be triggered after the successful reception of a frame. 1h (R/W) = IntPnd will be triggered after the successful reception of a frame.
9	RmtEn	R	0h	Remote enable 0h (R/W) = At the reception of a remote frame, TxRqst is not changed. 1h (R/W) = At the reception of a remote frame, TxRqst is set.
8	TxRqst	R	0h	Transmit Request 0h (R/W) = This message object is not waiting for a transmission. 1h (R/W) = The transmission of this message object is requested and is not yet done.
7	EoB	R	0h	Data frame has 0 to 8 data bits. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. 0h (R/W) = Data frame has 8 data bytes. 1h (R/W) = Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
6-4	RESERVED	R	0h	
3-0	DLC	R	0h	Data Length Code. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. 8h (R/W) = Data frame has 0 8 bits. 15h (R/W) = Data frame has 8 data bytes.

21.17.85 IF3DATA Register (Offset = 150h) [reset = 0h]

IF3DATA is shown in Figure 21-104 and described in Table 21-92.

Return to Summary Table.

IF3 Data A Register The data bytes of CAN messages are stored in the IF3 registers in the following order. In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 21-104. IF3DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data_3								Data_2								Data_1								Data_0							
R-0h								R-0h								R-0h								R-0h							

Table 21-92. IF3DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Data_3	R	0h	Data 3.
23-16	Data_2	R	0h	Data 2.
15-8	Data_1	R	0h	Data 1.
7-0	Data_0	R	0h	Data 0.

21.17.86 IF3DATB Register (Offset = 154h) [reset = 0h]

IF3DATB is shown in Figure 21-105 and described in Table 21-93.

Return to Summary Table.

IF3 Data B Register The data bytes of CAN messages are stored in the IF3 registers in the following order. In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

Figure 21-105. IF3DATB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data_7								Data_6								Data_5								Data_4							
R-0h								R-0h								R-0h								R-0h							

Table 21-93. IF3DATB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Data_7	R	0h	Data 7.
23-16	Data_6	R	0h	Data 6.
15-8	Data_5	R	0h	Data 5.
7-0	Data_4	R	0h	Data 4.

21.17.87 Reserved_13 Register (Offset = 158h) [reset = 0h]

Reserved_13 is shown in Figure 21-106 and described in Table 21-94.

Return to Summary Table.

Figure 21-106. Reserved_13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-94. Reserved_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.88 Reserved_14 Register (Offset = 15Ch) [reset = 0h]

Reserved_14 is shown in Figure 21-107 and described in Table 21-95.

Return to Summary Table.

Figure 21-107. Reserved_14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-95. Reserved_14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.89 IF3UPD12 Register (Offset = 160h) [reset = 0h]

IF3UPD12 is shown in Figure 21-108 and described in Table 21-96.

Return to Summary Table.

IF3 Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 21-108. IF3UPD12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UpdEn_1																IF3UpdEn_0															
R/W-0h																R/W-0h															

Table 21-96. IF3UPD12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IF3UpdEn_1	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.
15-0	IF3UpdEn_0	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

21.17.90 IF3UPD34 Register (Offset = 164h) [reset = 0h]

IF3UPD34 is shown in Figure 21-109 and described in Table 21-97.

Return to Summary Table.

IF3 Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 21-109. IF3UPD34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UpdEn_1																IF3UpdEn_0															
R/W-0h																R/W-0h															

Table 21-97. IF3UPD34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IF3UpdEn_1	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.
15-0	IF3UpdEn_0	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

21.17.91 IF3UPD56 Register (Offset = 168h) [reset = 0h]

IF3UPD56 is shown in Figure 21-110 and described in Table 21-98.

Return to Summary Table.

IF3 Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 21-110. IF3UPD56 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UpdEn_1																IF3UpdEn_0															
R/W-0h																R/W-0h															

Table 21-98. IF3UPD56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IF3UpdEn_1	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.
15-0	IF3UpdEn_0	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

21.17.92 IF3UPD78 Register (Offset = 16Ch) [reset = 0h]

IF3UPD78 is shown in Figure 21-111 and described in Table 21-99.

Return to Summary Table.

IF3 Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set. IF3 Update enable should not be set for transmit objects.

Figure 21-111. IF3UPD78 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UpdEn_1																IF3UpdEn_0															
R/W-0h																R/W-0h															

Table 21-99. IF3UPD78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IF3UpdEn_1	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.
15-0	IF3UpdEn_0	R/W	0h	IF3 Update Enabled (for all message objects) 0h (R/W) = Automatic IF3 update is disabled for this message object. 1h (R/W) = Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

21.17.93 Reserved_15 Register (Offset = 170h) [reset = 0h]

Reserved_15 is shown in Figure 21-112 and described in Table 21-100.

Return to Summary Table.

Figure 21-112. Reserved_15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-100. Reserved_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.94 Reserved_16 Register (Offset = 174h) [reset = 0h]

Reserved_16 is shown in Figure 21-113 and described in Table 21-101.

Return to Summary Table.

Figure 21-113. Reserved_16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-101. Reserved_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.95 Reserved_17 Register (Offset = 178h) [reset = 0h]

Reserved_17 is shown in Figure 21-114 and described in Table 21-102.

Return to Summary Table.

Figure 21-114. Reserved_17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-102. Reserved_17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.96 Reserved_18 Register (Offset = 17Ch) [reset = 0h]

Reserved_18 is shown in Figure 21-115 and described in Table 21-103.

Return to Summary Table.

Figure 21-115. Reserved_18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-103. Reserved_18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.97 Reserved_19 Register (Offset = 180h) [reset = 0h]

Reserved_19 is shown in Figure 21-116 and described in Table 21-104.

Return to Summary Table.

Figure 21-116. Reserved_19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-104. Reserved_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.98 Reserved_20 Register (Offset = 184h) [reset = 0h]

Reserved_20 is shown in Figure 21-117 and described in Table 21-105.

Return to Summary Table.

Figure 21-117. Reserved_20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-105. Reserved_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.99 Reserved_21 Register (Offset = 188h) [reset = 0h]

Reserved_21 is shown in Figure 21-118 and described in Table 21-106.

Return to Summary Table.

Figure 21-118. Reserved_21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-106. Reserved_21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.100 Reserved_22 Register (Offset = 18Ch) [reset = 0h]

Reserved_22 is shown in Figure 21-119 and described in Table 21-107.

Return to Summary Table.

Figure 21-119. Reserved_22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-107. Reserved_22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.101 Reserved_23 Register (Offset = 190h) [reset = 0h]

Reserved_23 is shown in Figure 21-120 and described in Table 21-108.

Return to Summary Table.

Figure 21-120. Reserved_23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-108. Reserved_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.102 Reserved_24 Register (Offset = 194h) [reset = 0h]

Reserved_24 is shown in Figure 21-121 and described in Table 21-109.

Return to Summary Table.

Figure 21-121. Reserved_24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-109. Reserved_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.103 Reserved_25 Register (Offset = 198h) [reset = 0h]

Reserved_25 is shown in Figure 21-122 and described in Table 21-110.

Return to Summary Table.

Figure 21-122. Reserved_25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-110. Reserved_25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.104 Reserved_26 Register (Offset = 19Ch) [reset = 0h]

Reserved_26 is shown in Figure 21-123 and described in Table 21-111.

Return to Summary Table.

Figure 21-123. Reserved_26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-111. Reserved_26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.105 Reserved_27 Register (Offset = 1A0h) [reset = 0h]

Reserved_27 is shown in Figure 21-124 and described in Table 21-112.

Return to Summary Table.

Figure 21-124. Reserved_27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-112. Reserved_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.106 Reserved_28 Register (Offset = 1A4h) [reset = 0h]

Reserved_28 is shown in Figure 21-125 and described in Table 21-113.

Return to Summary Table.

Figure 21-125. Reserved_28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-113. Reserved_28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.107 Reserved_29 Register (Offset = 1A8h) [reset = 0h]

Reserved_29 is shown in Figure 21-126 and described in Table 21-114.

Return to Summary Table.

Figure 21-126. Reserved_29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-114. Reserved_29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.108 Reserved_30 Register (Offset = 1ACh) [reset = 0h]

Reserved_30 is shown in Figure 21-127 and described in Table 21-115.

Return to Summary Table.

Figure 21-127. Reserved_30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-115. Reserved_30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.109 Reserved_31 Register (Offset = 1B0h) [reset = 0h]

Reserved_31 is shown in Figure 21-128 and described in Table 21-116.

Return to Summary Table.

Figure 21-128. Reserved_31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-116. Reserved_31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.110 Reserved_32 Register (Offset = 1B4h) [reset = 0h]

Reserved_32 is shown in Figure 21-129 and described in Table 21-117.

Return to Summary Table.

Figure 21-129. Reserved_32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-117. Reserved_32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.111 Reserved_33 Register (Offset = 1B8h) [reset = 0h]

Reserved_33 is shown in Figure 21-130 and described in Table 21-118.

Return to Summary Table.

Figure 21-130. Reserved_33 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-118. Reserved_33 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.112 Reserved_34 Register (Offset = 1BCh) [reset = 0h]

Reserved_34 is shown in Figure 21-131 and described in Table 21-119.

Return to Summary Table.

Figure 21-131. Reserved_34 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-119. Reserved_34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.113 Reserved_35 Register (Offset = 1C0h) [reset = 0h]

Reserved_35 is shown in Figure 21-132 and described in Table 21-120.

Return to Summary Table.

Figure 21-132. Reserved_35 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-120. Reserved_35 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.114 Reserved_36 Register (Offset = 1C4h) [reset = 0h]

Reserved_36 is shown in Figure 21-133 and described in Table 21-121.

Return to Summary Table.

Figure 21-133. Reserved_36 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-121. Reserved_36 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.115 Reserved_37 Register (Offset = 1C8h) [reset = 0h]

Reserved_37 is shown in Figure 21-134 and described in Table 21-122.

Return to Summary Table.

Figure 21-134. Reserved_37 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-122. Reserved_37 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.116 Reserved_38 Register (Offset = 1CCh) [reset = 0h]

Reserved_38 is shown in Figure 21-135 and described in Table 21-123.

Return to Summary Table.

Figure 21-135. Reserved_38 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-123. Reserved_38 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.117 Reserved_39 Register (Offset = 1D0h) [reset = 0h]

Reserved_39 is shown in Figure 21-136 and described in Table 21-124.

Return to Summary Table.

Figure 21-136. Reserved_39 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-124. Reserved_39 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.118 Reserved_40 Register (Offset = 1D4h) [reset = 0h]

Reserved_40 is shown in Figure 21-137 and described in Table 21-125.

Return to Summary Table.

Figure 21-137. Reserved_40 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-125. Reserved_40 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.119 Reserved_41 Register (Offset = 1D8h) [reset = 0h]

Reserved_41 is shown in Figure 21-138 and described in Table 21-126.

Return to Summary Table.

Figure 21-138. Reserved_41 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-126. Reserved_41 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.120 Reserved_42 Register (Offset = 1DCh) [reset = 0h]

Reserved_42 is shown in Figure 21-139 and described in Table 21-127.

Return to Summary Table.

Figure 21-139. Reserved_42 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-																															

Table 21-127. Reserved_42 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0	

21.17.121 TIOC Register (Offset = 1E0h) [reset = 0h]

TIOC is shown in Figure 21-140 and described in Table 21-128.

Return to Summary Table.

CAN TX IO Control Register The CAN_TX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are only writable if Init bit of the CAN control register is set. The OD, Func, Dir and Out bits of the CAN TX IO control register are forced to certain values when Init bit of CAN control register is reset (see bit descriptions).

Figure 21-140. TIOC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					PU	PD	OD
R-0h					R/W-0h	R/W-0h	R/WP-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				Func	Dir	Out	In
R-0h				R/WP-0h	R/WP-0h	R/WP-0h	R-0h

Table 21-128. TIOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18	PU	R/W	0h	CAN_TX pull up/pull down select. This bit is only active when CAN_TX is configured to be an input. 0h (R/W) = CAN_TX pull down is selected, when pull logic is active (PD = 0). 1h (R/W) = CAN_TX pull up is selected, when pull logic is active (PD = 0).
17	PD	R/W	0h	CAN_TX pull disable. This bit is only active when CAN_TX is configured to be an input. 0h (R/W) = CAN_TX pull is active 1h (R/W) = CAN_TX pull is disabled
16	OD	R/WP	0h	CAN_TX open drain enable. This bit is only active when CAN_TX is configured to be in GIO mode (TIOC.Func=0). Forced to '0' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_TX pin is configured in push/pull mode. 1h (R/W) = The CAN_TX pin is configured in open drain mode.
15-4	RESERVED	R	0h	
3	Func	R/WP	0h	CAN_TX function. This bit changes the function of the CAN_TX pin. Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = CAN_TX pin is in GIO mode. 1h (R/W) = CAN_TX pin is in functional mode (as an output to transmit CAN data).
2	Dir	R/WP	0h	CAN_TX data direction. This bit controls the direction of the CAN_TX pin when it is configured to be in GIO mode only (TIOC.Func=0). Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_TX pin is an input. 1h (R/W) = The CAN_TX pin is an output

Table 21-128. TIOC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	Out	R/WP	0h	CAN_TX data out write. This bit is only active when CAN_TX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_TX pin. Forced to Tx output of the CAN core, if Init bit of CAN control register is reset. 0h (R/W) = The CAN_TX pin is driven to logic low 1h (R/W) = The CAN_TX pin is driven to logic high
0	In	R	0h	CAN_TX data in. Note: When CAN_TX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module). 0h (R/W) = The CAN_TX pin is at logic low 1h (R/W) = The CAN_TX pin is at logic high

21.17.122 RIOC Register (Offset = 1E4h) [reset = 0h]

RIOC is shown in Figure 21-141 and described in Table 21-129.

Return to Summary Table.

CAN RX IO Control Register The CAN_RX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are writable only if Init bit of CAN control register is set. The OD, Func and Dir bits of the CAN RX IO control register are forced to certain values when the Init bit of CAN control register is reset (see bit descriptions).

Figure 21-141. RIOC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					PU	PD	OD
R-0h					R/W-0h	R/W-0h	R/WP-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				Func	Dir	Out	In
R-0h				R/WP-0h	R/WP-0h	R/WP-0h	R-0h

Table 21-129. RIOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18	PU	R/W	0h	CAN_RX pull up/pull down select. This bit is only active when CAN_RX is configured to be an input. 0h (R/W) = CAN_RX pull down is selected, when pull logic is active (PD = 0). 1h (R/W) = CAN_T=RX pull up is selected, when pull logic is active(PD = 0).
17	PD	R/W	0h	CAN_RX pull disable. This bit is only active when CAN_TX is configured to be an input. 0h (R/W) = CAN_RX pull is active 1h (R/W) = CAN_RX pull is disabled
16	OD	R/WP	0h	CAN_RX open drain enable. This bit is only active when CAN_RX is configured to be in GIO mode (TIOC.Func=0). Forced to '0' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_RX pin is configured in push/pull mode. 1h (R/W) = The CAN_RX pin is configured in open drain mode.
15-4	RESERVED	R	0h	
3	Func	R/WP	0h	CAN_RX function. This bit changes the function of the CAN_RX pin. Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = CAN_RX pin is in GIO mode. 1h (R/W) = CAN_RX pin is in functional mode (as an output to transmit CAN data).
2	Dir	R/WP	0h	CAN_RX data direction. This bit controls the direction of the CAN_RX pin when it is configured to be in GIO mode only (TIOC.Func=0). Forced to '1' if Init bit of CAN control register is reset. 0h (R/W) = The CAN_RX pin is an input. 1h (R/W) = The CAN_RX pin is an output

Table 21-129. RIOC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	Out	R/WP	0h	CAN_RX data out write. This bit is only active when CAN_RX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_RX pin. Forced to Tx output of the CAN core, if Init bit of CAN control register is reset. 0h (R/W) = The CAN_RX pin is driven to logic low 1h (R/W) = The CAN_RX pin is driven to logic high
0	In	R	0h	CAN_RX data in. Note: When CAN_RX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (for example, while reset of the DCAN module). 0h (R/W) = The CAN_RX pin is at logic low 1h (R/W) = The CAN_RX pin is at logic high

Modular Controller Area Network (MCAN)

This section describes the Modular Controller Area Network (MCAN).

Topic **Page**

22.1 MCAN (14xx,16xx,18xx) 2600
22.2 Modular Controller Area Network (68xx MCAN) 2733

22.1 MCAN (14xx,16xx,18xx)

22.1.1 MCAN Overview

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time control with a high level of security. CAN has high immunity to electrical interference and the ability to self-diagnose and repair data errors. In a CAN network, many short messages are broadcast to the entire network, which provides for data consistency in every node of the system.

The MCAN module supports both classic CAN and CAN FD (CAN with Flexible Data-Rate) specifications. CAN FD feature allows high throughput and increased payload per data frame. The classic CAN and CAN FD devices can coexist on the same network without any conflict.

The device supports one MCAN module connecting to the CAN network through external (for the device) transceiver for connection to the physical layer. The MCAN module supports up to 10 Mbit/s data rate and is compliant to the CAN protocol specification 2.0 part A, B and ISO 11898-1.

22.1.1.1 Features

The MCAN module implements the following features:

- Conforms with CAN Protocol 2.0 A, B and ISO 11898-1
- Full CAN FD support (up to 64 data bytes)
- AUTOSAR and SAE J1939 support
- Up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO, up to 32 elements
- Configurable Transmit Queue, up to 32 elements
- Configurable Transmit Event FIFO, up to 32 elements
- Up to 64 dedicated Receive Buffers
- Two configurable Receive FIFOs, up to 64 elements each
- Up to 128 filter elements
- Internal Loopback mode for self-test
- Maskable interrupts, two interrupt lines
- Two clock domains (CAN clock / Host clock)
- Parity / ECC support - Message RAM single error correction and double error detection (SECDED) mechanism
- Local power-down and wakeup support
- Timestamp Counter

Not supported features:

- Full Message Memory capacity (4352 words). Only 1600 words implemented.
- Debug on CAN (Debug DMA)
- Host bus read and write bursts
- Host bus firewall
- GPIO mode
- External (IO) Loopback mode
- Device clock domains monitoring (using DCC module)

22.1.2 MCAN Functional Description

The MCAN module performs CAN protocol communication according to the CAN protocol Specification 2.0 part A, B and ISO 11898-1. The bit rate can be programmed to values up to 10 Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

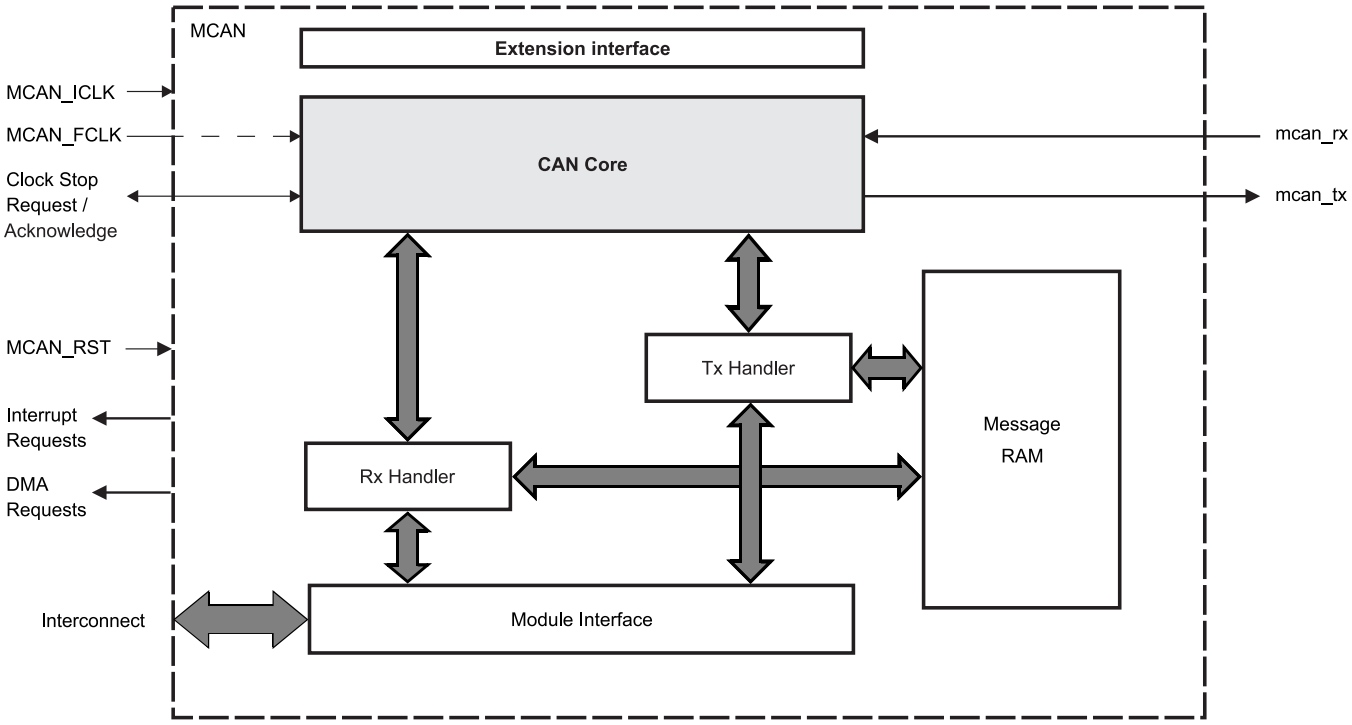
For communication on a CAN network, individual message frames can be configured. The message frames and identifier masks are stored in the Message RAM.

All functions concerning the handling of messages are implemented in the Message Handler.

The register set of the MCAN module can be accessed directly via the module interface. These registers are used to control and configure the CAN core and the Message Handler, and to access the Message RAM.

Figure 22-1 shows the MCAN module block diagram.

Figure 22-1. MCAN Block Diagram



The MCAN module blocks description:

- **CAN Core**: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1 protocol functions and supports 11-bit and 29-bit identifiers.
- **Tx Handler**: Controls the message transfer from the external Message RAM to the CAN Core. A maximum of 32 Tx Buffers can be configured for transmission. Tx buffers can be used as dedicated Tx Buffers, as Tx FIFO, part of a Tx Queue, or as a combination of them.
- **Rx Handler**: Controls the transfer of received messages from the CAN Core to the external Message RAM. The Rx Handler supports two Receive FIFOs, each of configurable size, and up to 64 dedicated Rx Buffers for storage of all messages that have passed acceptance filtering. A dedicated Rx Buffer, in contrast to a Receive FIFO, is used to store only messages with a specific identifier.
- **Message RAM**: the main purpose of the Message RAM is to store Rx/Tx messages, Tx Event elements, and Message ID Filter elements (for more information, see Section 22.1.2.11, *Message RAM*).
- **Module Interface**: The MCAN module registers are accessed by the user software through a 32-bit peripheral bus interface.
- **Clocking**: Two clocks are provided to the MCAN module: the peripheral synchronous clock (interface clock - MCAN_ICLK) and the peripheral asynchronous clock (functional clock - MCAN_FCLK).

- **Extension Interface**: All flags from the Interrupt Register (MCAN_IR) as well as selected internal status and control signals are routed to this interface.

22.1.2.1 Module Clocking Requirements

Two clocks are provided to the MCAN module:

- the peripheral synchronous clock (MCAN_ICLK) as the general module clock source
- and the peripheral asynchronous clock (MCAN_FCLK) provided to the CAN core for generating the CAN bit timing.

Within the MCAN module there is a synchronization mechanism implemented to ensure safe data transfer between the two clock domains. There are synchronization between the signals from the Host clock domain to the CAN clock domain and vice versa and between the reset signal (MCAN_RST) to the Host clock domain and to the CAN clock domain.

NOTE: MCAN_ICLK must always be higher or equal to MCAN_FCLK, in order to achieve a stable functionality of the MCAN module. Here, also the frequency shift of the modulated MCAN_ICLK has to be considered:

$f_{0,\ ICLK}(OCP) \pm \Delta f_{FM,\ ICLK}(OCP) \geq f_{FCLK}$

For more information on how to configure the relevant clock source registers, see , *PRCM* and the device data manual.

22.1.2.2 Interrupt and DMA Requests

The MCAN module provides interrupt and DMA requests. They are configured via the Host CPU. The Suspend Mode prevents the interrupt and DMA requests from propagating to the Host CPU (for more information, see Section 22.1.2.4.8.2, *Suspend Mode*).

22.1.2.2.1 Interrupt Requests

The MCAN module has two interrupt lines. The first interrupt line (INT0) is associated with the MCAN core. There are 30 internal interrupt sources. The interrupts are 'level high' interrupts.

For more information, see the following registers:

- Interrupt Register (MCAN_IR)
- Interrupt Enable (MCAN_IE)
- Interrupt Line Select (MCAN_ILS)
- Interrupt Line Enable (MCAN_ILE)

The MCAN module is capable of issuing an ECC interrupt. After clearing the ECC interrupt source, the application software must also write '1' to MCANSS_ECC_EOI[8] ECC_EOI bit (for more information, see Section 22.1.2.7.2, *ECC Aggregator*).

The second interrupt line (INT1) is associated with the External Timestamp Counter. When the External Timestamp Counter rolls over it produces an interrupt (see Section 22.1.2.5.1, *External Timestamp Counter*).

For more information, see the following registers:

- Interrupt Clear Shadow Register (MCANSS_ICS)
- Interrupt Raw Status Register (MCANSS_IRS)
- Interrupt Enable Clear Shadow Register (MCANSS_IECS)
- Interrupt Enable Register (MCANSS_IE)
- Interrupt Enable Status (MCANSS_IES)
- End Of Interrupt (MCANSS_EOI)
- External Timestamp Prescaler (MCANSS_EXT_TS_PRESCALER)
- External Timestamp Unserviced Interrupts Counter (MCANSS_EXT_TS_UNSERVICED_INTR_CNTR)

22.1.2.2.2 DMA Requests

Functional transmit and Filter DMA requests are generated by the MCAN module based on the signaling in the Extention Interface. The DMA signaling uses a simple DMA request active high pulse. Only one Tx DMA event is provided by the MCAN module.

Standard and Extended message filters can be set to issue a pulse when a filter match occurs. These 'Filter Events' can be used to DMA messages from the Rx FIFO. The events are high level single clock cycle (MCAN_ICLK) pulses. Only two Filter DMA events are provided by the MCAN module.

22.1.2.3 Fuseable CAN FD Operation Enable

The Flexible Datarate feature of the MCAN module can be enabled by writing '1' to MCAN_CCCR[8] FDOE bit. A value of '0' on the primary configuration port (mcanss_enable_fdoe) will force the MCAN_CCCR[8] FDOE bit during write to the MCAN_CCCR register which will prevent the device from enabling and using the CAN FD mode.

22.1.2.4 Operating Modes

22.1.2.4.1 Software Initialization

Setting the MCAN_CCCR[0] INIT bit to '1' starts a software initialization. This is done either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off state. While the MCAN_CCCR[0] INIT bit is set, the message transfer is stopped and the status of the output mcan_tx pin is recessive (high). The counters of the Error Management Logic (EML) are unchanged. Setting the MCAN_CCCR[0] INIT bit does not change any configuration register. Resetting the MCAN_CCCR[0] INIT bit finishes the software initialization. After waiting for the occurrence of a sequence of 11 consecutive recessive bits (indication for Bus_Idle state) the message transfer starts.

Access to the MCAN configuration registers is only enabled when both MCAN_CCCR[0] INIT and MCAN_CCCR[1] CCE bits are set (write protection).

The MCAN_CCCR[1] CCE bit can only be set/reset while the MCAN_CCCR[0] INIT = '1'. The MCAN_CCCR[1] CCE bit is automatically reset when the MCAN_CCCR[0] INIT bit is reset.

The following registers are reset when the MCAN_CCCR[1] CCE bit is set:

- MCAN_HPMS - High Priority Message Status
- MCAN_RXF0S - Rx FIFO 0 Status
- MCAN_RXF1S - Rx FIFO 1 Status
- MCAN_TXFQS - Tx FIFO/Queue Status
- MCAN_TXBRP - Tx Buffer Request Pending
- MCAN_TXBTO - Tx Buffer Transmission Occurred
- MCAN_TXBCF - Tx Buffer Cancellation Finished
- MCAN_TXEFS - Tx Event FIFO Status

The Timeout Counter value MCAN_TOCV[15:0] TOC field is preset to the value configured by the MCAN_TOCC[31:16] TOP field when the MCAN_CCCR[1] CCE bit is set.

In addition the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR[1] CCE = '1'.

The following registers are only writeable while MCAN_CCCR[1] CCE = '0'

- MCAN_TXBAR - Tx Buffer Add Request
- MCAN_TXBCR - Tx Buffer Cancellation Request

MCAN_CCCR[7] TEST and MCAN_CCCR[5] MON bits can only be set by the Host CPU while MCAN_CCCR[0] INIT = '1' and MCAN_CCCR[1] CCE = '1'. Both bits may be reset at any time. The MCAN_CCCR[6] DAR bit can only be set/reset while MCAN_CCCR[0] INIT = '1' and MCAN_CCCR[1] CCE = '1'.

22.1.2.4.2 Normal Operation

Once the MCAN module is initialized and the MCAN_CCCR[0] INIT bit is reset to zero, the MCAN module synchronizes itself to the CAN bus and is ready for communication. After passing the acceptance filtering, received messages including Message Identifier (ID) and Data Length Code (DLC) are stored into a dedicated Rx Buffer or into Rx FIFO 0 / Rx FIFO 1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated.

NOTE: Automated transmission on reception of remote frames is not supported.

22.1.2.4.3 CAN FD Operation

There are two variants of CAN FD frame transmission:

- CAN FD frame transmission without bit rate switching
- CAN FD frame transmission where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame

In the CAN frames FDF = recessive (logical '1') signifies a CAN FD frame, FDF = dominant (logical '0') signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF - res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. Note that the coding of res = recessive is reserved for future expansion of the protocol. In case the MCAN module receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting the MCAN_PSR[14] EXE bit. When Protocol Exception Handling is enabled (MCAN_CCCR[12] PXHD = '0'), this causes the operation state to change from Receiver (MCAN_PSR[4:3] ACT = '10') to Integrating (MCAN_PSR[4:3] ACT = '00') at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR[12] PXHD = '1'), the MCAN will treat a recessive res bit as an form error and will respond with an error frame.

CAN FD operation is enabled by programming the MCAN_CCCR[8] FDOE bit. In case MCAN_CCCR[8] FDOE = '1', transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via the FDF bit in the respective Tx Buffer element.

With MCAN_CCCR[8] FDOE = '0', received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if the FDF bit of a Tx Buffer element is set. The MCAN_CCCR[8] FDOE and MCAN_CCCR[9] BRSE bits can only be changed while the MCAN_CCCR[0] INIT and MCAN_CCCR[1] CCE bits are both set. With MCAN_CCCR[8] FDOE = '0', the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format.

With MCAN_CCCR[8] FDOE = '1' and MCAN_CCCR[9] BRSE = '0', only FDF bit of a Tx Buffer element is evaluated. With MCAN_CCCR[8] FDOE = '1' and MCAN_CCCR[9] BRSE = '1', transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

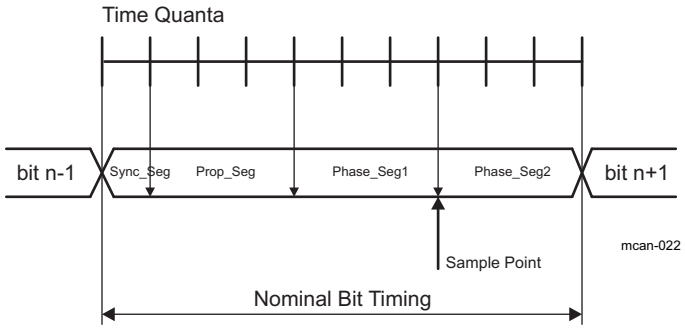
The coding of the DLC in the CAN FD format differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN (0 to 8 data bytes), the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to Table 22-1.

Table 22-1. DLC Coding in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

For CAN FD frames with bit rate switchingIn, the bit timing will be switched inside the frame after the BRS (Bit Rate Switch) bit in case this bit is recessive. In the CAN FD arbitration phase, before the BRS bit, the nominal CAN bit timing (see Figure 22-2) is used as configured by the Nominal Bit Timing and Prescaler Register MCAN_NBTP. In the following CAN FD data phase, the data phase bit timing is used as configured by the Data Bit Timing and Prescaler Register MCAN_DBTP. The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

Figure 22-2. CAN Bit Timing



The maximum configurable data phase bit timing depends on the CAN clock frequency (MCAN_FCLK). Example: with MCAN_FCLK = 20 MHz and the shortest configurable bit time of 4 t_q (time quanta), the bit rate in the data phase is 5 Mbit/s.

For both CAN FD without and CAN FD with bit rate switching the value of the ESI (Error Status Indicator) bit depends on transmitter's error state (see MCAN_PSR[11] RESI bit) monitored at the start of the transmission. If the transmitter has error passive flag the ESI bit is transmitted recessive, else it is transmitted dominant.

22.1.2.4.4 Transmitter Delay Compensation

22.1.2.4.4.1 Description

When only one CAN FD node is transmitting and all others are receivers the length of the bus line has no impact. When transmitting via the mcan_tx pin the MCAN module receives the transmitted data from its local CAN transceiver via the mcan_rx pin. The received data is delayed. If the transmitter delay is greater than TSEG1 (time segment before sample point), a bit error is detected.

The MCAN module provides a delay compensation mechanism to compensate the transmitter delay. The compensation mechanism enables transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Without transmitter delay compensation the bit rate in the data phase is limited by the transmitter delay.

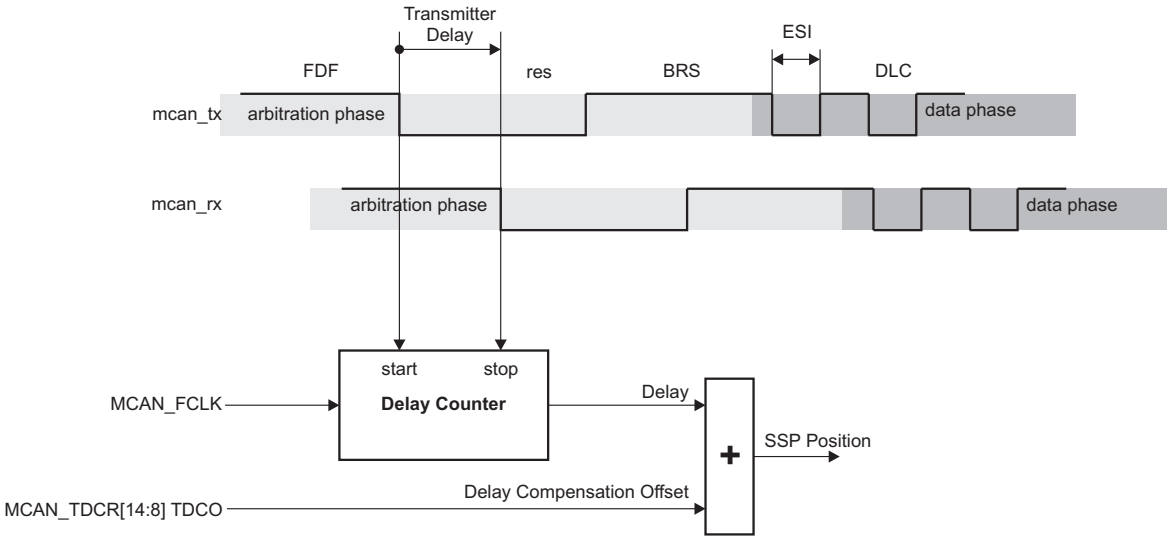
The mechanism enables configurations where the data bit time is shorter than the transmitter delay (it is described in detail in the new ISO11898-1). The transmitter delay compensation is enabled by setting the MCAN_DBTP[23] TDC bit to '1'.

The delayed transmit data is compared against the received data at the Secondary Sample Point (SSP) in order to check for bit errors during the data phase of transmitting nodes. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN's transmit output mcan_tx pin through the transceiver to the receive input mcan_rx pin plus the transmitter delay compensation offset configured by the MCAN_TDCR[14:8] TDCO field (see Figure 22-3). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (example: half of the bit time in the data phase). The position of the SSP is rounded down to the next integer number of mtq.

The actual transmitter delay compensation value can be checked by reading the MCAN_PSR[22:16] TDCV field. This filed is cleared when the MCAN_CCCR[0] INIT bit is set and is updated at each transmission of CAN FD frame while the MCAN_DBTP[23] TDC bit is set.

Figure 22-3. Transmitter Delay Measurement



22.1.2.4.4.2 Transmitter Delay Compensation Measurement

When transmitter delay compensation is enabled (by programming MCAN_DBTP[23] TDC = '1'), the measurement is started within each transmitted CAN FD frame at the falling edge of FDF bit to bit res. The measurement is stopped when this edge is seen at the receive input mcan_rx pin of the transmitter. The resolution of this measurement is one mtq (see Figure 22-3). The mtq (minimum time quantum) dimension is equal to the CAN clock period (MCAN_FCLK).

The use of a transmitter delay compensation filter window can be enabled by programming MCAN_TDCR[6:0] TDCF field. This filter feature defines a minimum value for the SSP position to avoid the case in which a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in an early taken SSP position. Dominant edges on the mcan_rx pin, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR[6:0] TDCF field and the mcan_rx pin is low.

The following boundary conditions have to be considered:

- The sum of the measured delay from the mcan_tx pin to the mcan_rx pin and the configured transmitter delay compensation offset (MCAN_TDCR[14:8] TDCO field) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from the mcan_tx pin to the mcan_rx pin and the configured transmitter delay compensation offset (MCAN_TDCR[14:8] TDCO) field has to be less or equal 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

22.1.2.4.5 Restricted Operation Mode

In Restricted Operation Mode the CAN node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The recieve and transmit error counters (MCAN_ECR[14:8] REC and MCAN_ECR[7:0] TEC) are frozen while CAN error logging (MCAN_ECR[23:16] CEL) is active. The Host CPU can set the MCAN module into Restricted Operation Mode by setting MCAN_CCCR[2] ASM bit. The bit can only be set by the Host CPU at any time when both MCAN_CCCR[2] CCE and MCAN_CCCR[1] INIT bits are set to '1'.

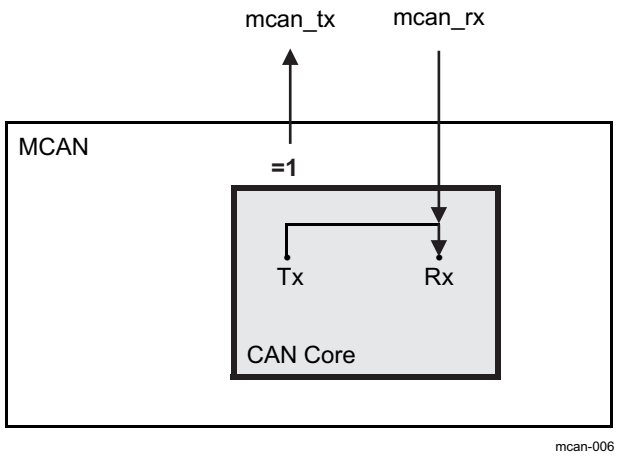
The Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the Host CPU has to reset MCAN_CCCR[2] ASM bit. This mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

NOTE: The Restricted Operation Mode must not be combined with the Loop Back Mode.

22.1.2.4.6 Bus Monitoring Mode

Entering Bus Monitoring Mode is done by setting the MCAN_CCCR[5] MON bit to '1'. In this mode (see ISO11898-1, *Bus Monitoring*), the MCAN module is able to receive valid data and remote frames, but cannot start a transmission. The MCAN module sends only recessive bits on the CAN bus. If the MCAN module is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN module monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode the MCAN_TXBRP register is held in reset state. The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 22-4 shows the connection of the mcan_tx and mcan_rx signals to the MCAN module in Bus Monitoring Mode.

Figure 22-4. Connection of Signals in Bus Monitoring Mode



22.1.2.4.7 Disabled Automatic Retransmission (DAR) Mode

According to the CAN Specification (see ISO11898-1, *Recovery Management*), the MCAN module provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled (see the MCAN_CCCR[6] DAR bit).

22.1.2.4.7.1 Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending MCAN_TXBRP[xx] TRPx bit is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

Successful transmission:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is not set

Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is set

Arbitration lost or frame transmission disturbed:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is not set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = '10' (transmission in spite of cancellation).

22.1.2.4.8 Power Down (Sleep Mode)

Entering Power Down mode is controlled via the input clock stop request signal (mcanss_clkstp_clkstop_req) or MCAN_CCCR[4] CSR bit. As long as the clock stop request signal is active, the MCAN_CCCR[4] CSR bit is read as '1'. When all pending transmission requests have completed, the MCAN module waits until bus idle state is detected. Then the MCAN module sets the MCAN_CCCR[1] INIT to '1' to prevent any further CAN transfers. The MCAN module acknowledges that it is ready for power down by setting the output clock stop acknowledge signal (mcanss_clkstop_clkstop_ack) to '1' and the MCAN_CCCR[3] CSA bit to '1'. In this state, before the clocks are switched off, further register accesses can be made. A write access to the MCAN_CCCR[1] INIT bit will have no effect. Now the module clock inputs MCAN_ICLK and MCAN_FCLK may be switched off.

To leave power down mode, the application has to turn on the module clocks before resetting the input clock stop request signal respectively the MCAN_CCCR[4] CSR flag bit. The MCAN will acknowledge this by resetting the output clock stop acknowledge signal respectively the MCAN_CCCR[3] CSA flag bit. Afterwards, the application can restart CAN communication by resetting MCAN_CCCR[1] INIT bit.

22.1.2.4.8.1 External Clock Stop Mode

The MCAN module supports two external clock stop modes:

- Immediate
- Graceful

In a graceful clock stop mode, when the clock stop request is asserted, the MCAN core will respond with clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. The MCAN_CCCR[0] INIT bit will be set, the MCAN core will go and stay Idle.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits to '1' (for more information, see Section 22.1.2.4.8.3, *Wakeup request*). When external clock stop request is removed and no suspend request is active, a read-modify-write to the MCAN_CCCR[0] INIT bit is performed to clear it.

22.1.2.4.8.2 Suspend Mode

The MCAN module supports two suspend modes:

- Immediate
- Graceful

In a graceful suspend mode (see the MCANSS_CTRL[3] FREE and MCANSS_CTRL[2] SOFT bits), when the suspend request is asserted, a clock stop request to the MCAN core is performed. The MCAN core will respond with clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. At that point the MCAN_CCCR[0] INIT bit will be set, the MCAN core will go and stay Idle. The suspend state can be verified by reading MCAN_CCCR[0] INIT bit.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits to '1' (for more information, see Section 22.1.2.4.8.3, *Wakeup request*). When suspend request is removed, if no external clock stop request is active, a read-modify-write to the MCAN_CCCR[0] INIT bit is performed to clear it.

During suspend mode the auto-clear feature is disabled. The following register fields have an auto-clear feature:

- MCAN_ECR[23:16] CEL
- MCAN_PSR[2:0] LEC
- MCAN_PSR[10:8] DLEC
- MCAN_PSR[11] RESI
- MCAN_PSR[12] RBRS
- MCAN_PSR[13] RFDF
- MCAN_PSR[14] PXE

22.1.2.4.8.3 Wakeup request

Issuing a clock stop request puts the MCAN module into Power Down mode (Sleep Mode). During transition from IDLE to ACTIVE, if the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits are enabled, after the MCAN Core respond to the removal of the clock stop request with removing the clock stop acknowledge, a read-modify-write will be issued to clear the MCAN_CCCR[0] INIT bit and the MCAN core will resume operation.

If the MCANSS_CTRL[4] WAKEUPREQEN bit is set, the MCAN module provides a wakeup request (SWakeup) on any of the following wakeup events:

- The receive mcan_rx pin is dominant (logical 0)
- OCP access is performed

To clear the SWakeup in case any of these events is active, the MCANSS_CTRL[4] WAKEUPREQEN bit should be cleared. The MCAN module adds a third wakeup event source - interrupt line 0 (INT0). In this case the SWakeup is cleared by clearing the interrupt source.

22.1.2.4.9 Test Modes

The MCAN_TEST register write access is enabled by setting the test mode enable MCAN_CCCR[7] TEST bit to '1'. The MCAN_TEST register allows the configuration of the test modes and test functions.

The CAN transmit mcan_tx pin has four output functions. One of those functions can be selected by programming the MCAN_TEST[6:5] TX filed. Additionally to its default function (the serial data output) it can drive the CAN Sample Point signal to monitor the MCAN's bit timing and it can drive constant dominant or recessive values.

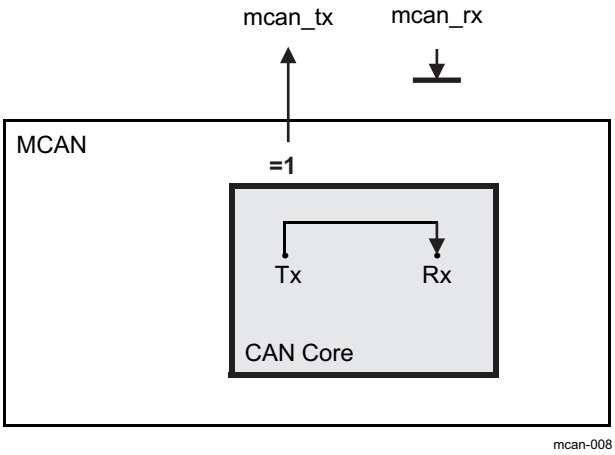
The actual value of the CAN receive mcan_rx pin can be monitored from MCAN_TEST[7] RX bit. Both functions can be used to check the CAN bus physical layer. Due to the synchronization mechanism between CAN clock (MCAN_FCLK) and Host clock (MCAN_ICLK) domain, there may be a delay of several Host clock periods between writing to the MCAN_TEST[6:5] TX filed until the new configuration is visible at the output mcan_tx pin. This applies also when reading input mcan_rx pin via the MCAN_TEST[7] RX bit.

NOTE: Test modes should be used for self test only. The software control for mcan_tx pin interferes with all CAN protocol functions. It is not recommended to use test modes for application.

22.1.2.4.9.1 Internal Loop Back Mode

The MCAN module can be set into Internal Loop Back Mode by programming MCAN_TEST[4] LBCK and MCAN_CCCR[5] MON bits to '1'. The Internal Loop Back Mode is used for a 'Hot Selftest'. The 'Hot Selftest' allows the MCAN module to be tested without affecting a running CAN system connected to the mcan_tx and mcan_rx pins. In this mode mcan_rx pin is disconnected from the MCAN module and mcan_tx pin is held recessive. Figure 22-5 shows the connection of the mcan_tx and mcan_rx pins to the MCAN module in case of Internal Loop Back Mode.

Figure 22-5. Internal Loop Back Mode



22.1.2.5 Timestamp Generation

The MCAN module has integrated a 16-bit wrap-around counter for timestamp generation. The timestamp counter prescaler MCAN_TSCC[19:16] TCP field can be configured to clock the counter in multiples of CAN bit times (1-16). The counter is readable via the MCAN_TSCV[15:0] TSC field. A write access to the MCAN_TSCV register resets the counter to zero. When the timestamp counter wraps around the interrupt MCAN_IR[16] TSW flag is set. On start of a frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element. For more information, see Section 22.1.2.11, *Message RAM*.

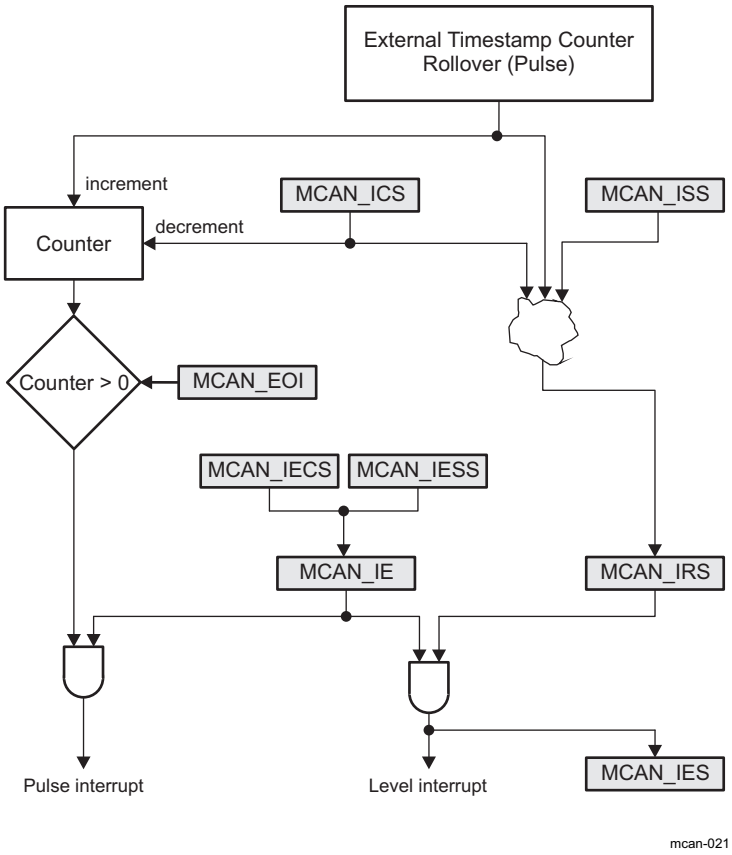
22.1.2.5.1 External Timestamp Counter

For CAN FD operation mode the MCAN core requires an External Timestamp Counter. An externally generated 16-bit vector may substitute the integrated 16-bit CAN bit time counter (internal timestamp counter) for receive and transmit timestamp generation. An external 16-bit timestamp counter can be used by programming the MCAN_TSCC[1:0] TSS field.

The External Timestamp Counter uses the interface clock (MCAN_ICLK) as a reference clock. The MCAN Core accepts a 16-bit timestamp. A 24-bit prescaler provides a programmable resolution for the timestamp (see MCANSS_EXT_TS_PRESCALER[23:0] PRESCALER field). When disabled the counter is reset back to zero. While enabled the counter keeps incrementing. When the timestamp rolls over the MCAN_IRQ_TS interrupt is generated. The MCAN module provides both pulse and level interrupt type for this interrupt.

When the timestamp rolls over the MCANSS_IRS register is set (see Figure 22-6). The MCANSS_IE register can be affected by writing to the MCAN_IESS register to set or to the MCANSS_IECS register to clear. The level interrupt is a reflection of both MCANSS_IRS and MCANSS_IE being set. The MCANSS_IES register reflects the level interrupt. When an rollover event occurs the interrupt counter is incremented. Writing to the MCANSS_ICS register to clear the MCANSS_IRS register will also decrement the interrupt counter. Writing to the MCANSS_EOI register will issue another pulse if the interrupt counter is not zero.

Figure 22-6. External Timestamp Counter Interrupt



22.1.2.6 Timeout Counter

The MCAN module has integrated a 16-bit Timeout Counter. It is used to signal timeout conditions for the Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO Message RAM elements. The Timeout Counter is configured via the MCAN_TOCC register. It is enabled via the MCAN_TOCC[0] ETOC bit. The Timeout Counter operates as down-counter and uses the same prescaler programmed by the MCAN_TSCC[19:16] TCP field as the Timestamp Counter. The actual counter value can be monitored from the MCAN_TOCV[15:0] TOC field. The Timeout Counter can be started only when MCAN_CCCR[1] INIT = '0' and stopped when MCAN_CCCR[1] INIT = '1' (example: when the MCAN enters Bus_Off state). The operation mode is selected by the MCAN_TOCC[2:1] TOS field. When Continuous Mode is selected, the counter starts when MCAN_TOCV[1] INIT = '0', a write to the MCAN_TOCV register presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field and continues down-counting.

In case the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field. Down-counting is started when the first FIFO element is stored. Writing to the MCAN_TOCV register has no effect. When the counter reaches zero, the interrupt MCAN_IR[18] TOO flag is set.

In Continuous Mode, the counter is immediately restarted at the value configured by the MCAN_TOCC[31:16] TOP field.

22.1.2.7 Safety

The Message Memory is wrapped in an ECC wrapper provididing SECDED parity functionalilty. The ECC wrapper is controlled by an ECC Aggregator.

22.1.2.7.1 ECC Wrapper

The ECC wrapper provides Single Error Correction (SEC) and Double Error Detection (DED) parity to the Message Memory content. It has side band signals for error notification. The ECC Wrapper implements an error injection test mode.

The error correction is done using a lazy write back. When an error is detected, it is noted in a FIFO Queue which waits for an access gap to write the data back and refresh the memory. If a transaction writes new data to the compromised entry before the lazy write back completes, the write back is discarded.

22.1.2.7.2 ECC Aggregator

This section describes the functional details of the ECC Aggregator module.

22.1.2.7.2.1 ECC Aggregator Overview

The ECC Aggregator module supports the following general features:

- Provides a mechanism to control and monitor the ECC RAM in the MCAN module.
- Provides software access to all the ECC related registers.
- Supports software readable status of ECC single/double-bit errors and associated info such as RAM address and data bit(s) that are in error.
- Aggregates level pending status from the ECC RAM into a single interrupt to the Host CPU.

The following feature is not supported:

- Statistics such as tracking the number of single and double-bit errors. If needed, these operations can be handled by software.

22.1.2.7.2.2 ECC Aggregator Registers

There are 3 groups of registers in the ECC aggregator module:

- Global registers - Aggregator Revision Register (MCANSS_ECC_AGGR_REVISION), ECC Vector Register (MCANSS_ECC_VECTOR), Misc Status Register (MCANSS_ECC_MISC_STATUS), ECC Control Register (MCANSS_ECC_CONTROL), and ECC Wrapper Revision Register (MCANSS_ECC_WRAP_REVISION).
- Control and status registers - ECC Error Control Registers (MCANSS_ECC_ERR_CTRL1 and MCANSS_ECC_ERR_CTRL2) and ECC Error Status Registers (MCANSS_ECC_ERR_STAT1 and MCANSS_ECC_ERR_STAT2).
- Interrupt registers - interrupt status, interrupt enable set, interrupt enable clear and EOI (End Of Interrupt) registers that are part of a standard interrupt module. For more information, see the following registers:
  - MCANSS_ECC_SEC_EOI_REG
  - MCANSS_ECC_SEC_STATUS_REG0
  - MCANSS_ECC_SEC_ENABLE_SET_REG0
  - MCANSS_ECC_SEC_ENABLE_CLR_REG0
  - MCANSS_ECC_DED_EOI_REG
  - MCANSS_ECC_DED_STATUS_REG0
  - MCANSS_ECC_DED_ENABLE_SET_REG0
  - MCANSS_ECC_DED_ENABLE_CLR_REG0

22.1.2.7.2.3 Reads to ECC Control and Status Registers

The reads to the ECC control and status registers are triggered by writing a 'read message' to the ECC Vector Register as described below:

- Software writes value (the ECC RAM ID) to the MCANSS_ECC_VECTOR[10-0] ECC_VECTOR field to select the ECC RAM for control or status.

- Software writes '1' to the MCANSS_ECC_VECTOR[15] RD_SVBUS bit to trigger a read.
- Software writes read address to the MCANSS_ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field.
- Software then polls the MCANSS_ECC_VECTOR[24] RD_SVBUS_DONE bit to check if it is '1'. This bit indicates that the read operation has completed.
- Software reads the data from the ECC control or status register. The following clock cycle (MCAN_ICLK) returns the read data.

22.1.2.7.2.4 ECC Interrupts

The ECC aggregator module aggregates the level pending status from the ECC RAM into a single EOI-handshake based interrupt to the Host CPU. Software is expected to follow the sequence described below:

- Software enables the interrupts for the ECC RAM by writing to the MCANSS_ECC_SEC_ENABLE_SET_REG0 / MCANSS_ECC_DED_ENABLE_SET_REG0 register.
- Software writes the ECC RAM ID in the MCANSS_ECC_VECTOR[10-0] ECC_VECTOR.
- Software writes the MCANSS_ECC_VECTOR[15] RD_SVBUS bit to trigger the read.
- Software writes the MCANSS_ECC_ERR_STAT1 register address to the MCANSS_ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field. Software will need to load the 'read message' in the MCANSS_ECC_VECTOR register again if it needs to read the MCANSS_ECC_ERR_STAT2 register.
- Software polls the MCANSS_ECC_VECTOR[24] RD_SVBUS_DONE bit. When this bit is set, a read of the MCANSS_ECC_ERR_STAT1 / MCANSS_ECC_ERR_STAT2 register is performed.
- After the interrupt has been serviced, software will clear the interrupt status by writing to the MCANSS_ECC_ERR_STAT1[8] CLR_ECC_SEC or MCANSS_ECC_ERR_STAT1[9] CLR_ECC_DED bit depending on the type of the ECC error.
- Software has to poll the MCANSS_ECC_ERR_STAT1 register to guarantee that the status bit has been cleared.
- Software will write to the MCANSS_ECC_SEC_EOI_REG / MCANSS_ECC_DED_EOI_REG register to clear the interrupt.
- After clearing the ECC interrupt source, the application software must also write '1' to the MCANSS_ECC_EOI[8] ECC_EOI bit.

22.1.2.8 Rx Handling

The Rx Handler controls the following operations:

- Acceptance filtering
- The transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1)
- Rx FIFO Put and Get Index operations

22.1.2.8.1 Acceptance Filtering

The MCAN module is capable to configure two sets of acceptance filters - one set for standard and one set for extended identifiers. These filters can be assigned to an Rx Buffer or to one of the two Rx FIFOs.

The main features of the filter elements are:

- Each filter element can be configured as:
  - Range Filter (from - to)
  - Filter for specific IDs (for one or two dedicated IDs)
  - Classic Bit Mask Filter
- Each filter element can be enabled / disabled individually
- Each filter element can be configured for acceptance or rejection filtering
- Filters are checked sequentially and execution (acceptance filtering procedure) stops at the first matching filter element or when the end of the filter list is reached

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register
- Extended ID AND Mask (MCAN_XIDAM) register

Depending on the configuration of the filter element (see SFEC/EFEC in Section 22.1.2.11, *Message RAM*) if filter matches, one of the following actions is performed:

- Received frame is stored in FIFO 0 or FIFO 1
- Received frame is stored in Rx Buffer
- Received frame is stored in Rx Buffer and generation of pulse at filter event pin is performed. This is high level single MCAN_ICLK pulse. For more information, see Section 22.1.2.2.1, *DMA Requests*.
- Received frame is rejected
- Set High Priority Message interrupt flag MCAN_IR[8] HPM
- Set High Priority Message interrupt flag MCAN_IR[8] HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering starts when complete Message ID is received. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If a filter element matches - the Rx Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If an error condition occurs (for example: CRC error), this message is rejected with the following impact on the affected Rx Buffer or Rx FIFO:

- Rx Buffer:
  New Data flag (MCAN_NDAT1 / MCAN_NDAT2) of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data (for error type see MCAN_PSR[2:0] LEC respectively MCAN_PSR[10:8] DLEC fields).
- Rx FIFO:
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data (for error type see MCAN_PSR[2:0] LEC respectively MCAN_PSR[10:8] DLEC fields). If matching Rx FIFO is configured to operate in overwrite mode, the boundary conditions described in Section 22.1.2.8.2.2 have to be considered.

22.1.2.8.1.1 Range Filter

Each filter element can be configured to operate as Range Filter (Standard Filter Type SFT = '00' / Extended Filter Type EFT = '00'). The filter matches for all received message frames with IDs in the range from SFID1 to SFID2 (SFID2 ≥ SFID1) respectively in the range from EFID1 to EFID2 (EFID2 ≥ EFID1). For more information see Section 22.1.2.11.5, *Standard Message ID Filter Element* and Section 22.1.2.11.6, *Extended Message ID Filter Element*.

There are two options for range filtering of extended frames:

- Extended Filter Type EFT = '00': The Extended ID AND Mask (MCAN_XIDAM) is used for Range Filtering. The Message ID of received frames is ANDed with the Extended ID AND Mask (MCAN_XIDAM) before the range filter is applied.
- Extended Filter Type EFT = '11': The Extended ID AND Mask (MCAN_XIDAM) is not used for Range Filtering.

22.1.2.8.1.2 Filter for specific IDs

Each filter element can be configured to filter one or two dedicated Message IDs (Standard Filter Type SFT ='01' / Extended Filter Type EFT ='01'). To filter only one specific Message ID, the filter element has to be configured with SFID1 = SFID2 respectively EFID1 = EFID2. For more information see Section 22.1.2.11.5, *Standard Message ID Filter Element* and Section 22.1.2.11.6, *Extended Message ID Filter Element*.

22.1.2.8.1.3 Classic Bit Mask Filter

Classic bit mask filtering can filter groups of Message IDs (Standard Filter Type SFT ='10' / Extended Filter Type EFT ='10'). This is done by masking single bits of a received Message ID. In this case SFID1/EFID1 element is used as Message ID filter, while SFID2/EFID2 element is used as filter mask.

A '0' bit at the filter mask (SFID2/EFID2) will mask out the corresponding bit position of the configured Message ID filter (SFID1/EFID1) and the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are '1' are relevant for acceptance filtering.

There are two interesting cases:

- All mask bits are '1': a match occurs only when the received Message ID and the configured Message ID filter are identical.
- All mask bits are '0': all Message IDs match.

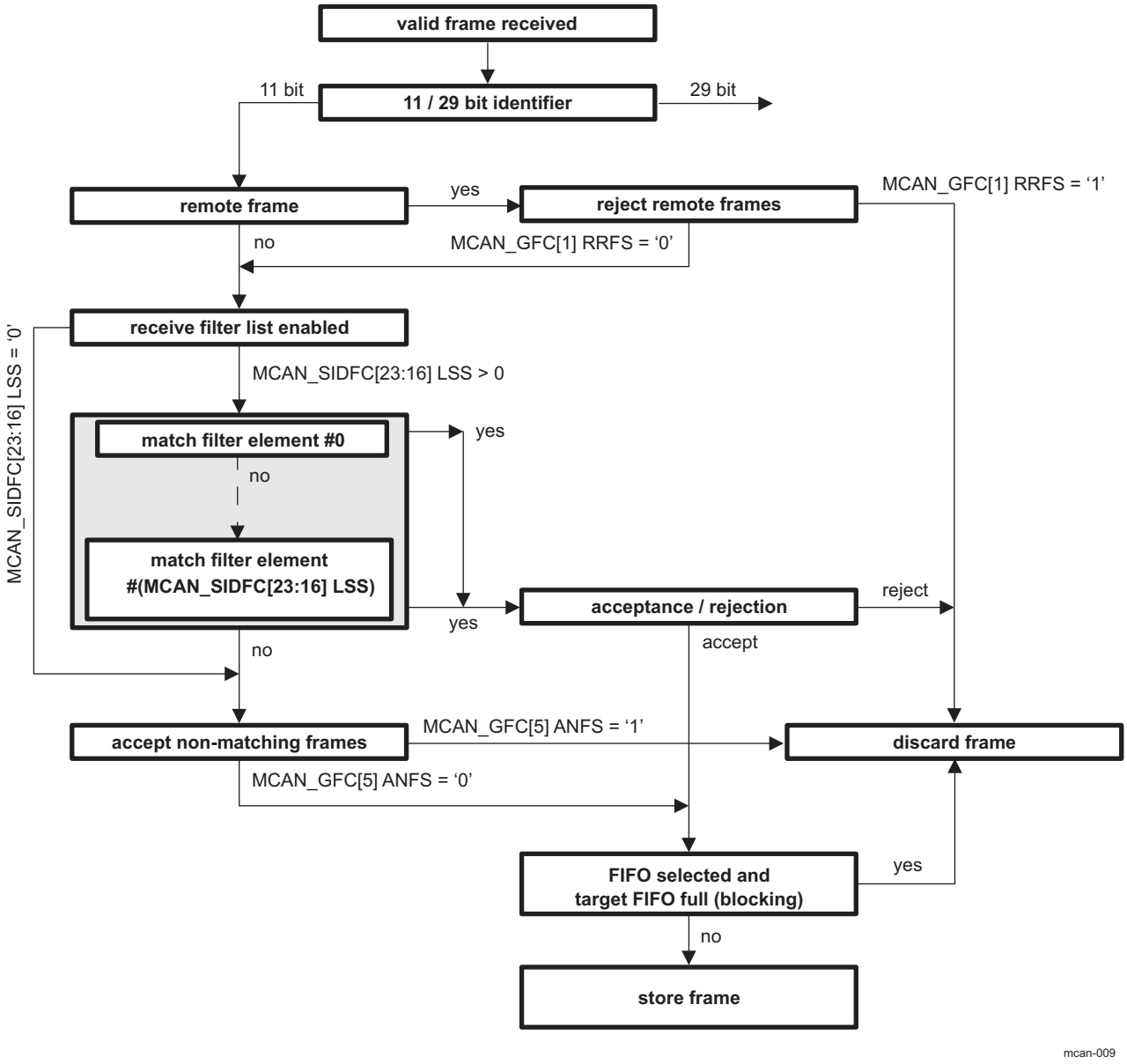
22.1.2.8.1.4 Standard Message ID Filtering

The standard Message ID (11-bit ID) filtering flow is shown in Figure 22-7. Section 22.1.2.11.5, *Standard Message ID Filter Element* describes the standard Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register

Figure 22-7. Standard Message ID Filter Path



22.1.2.8.1.5 Extended Message ID Filtering

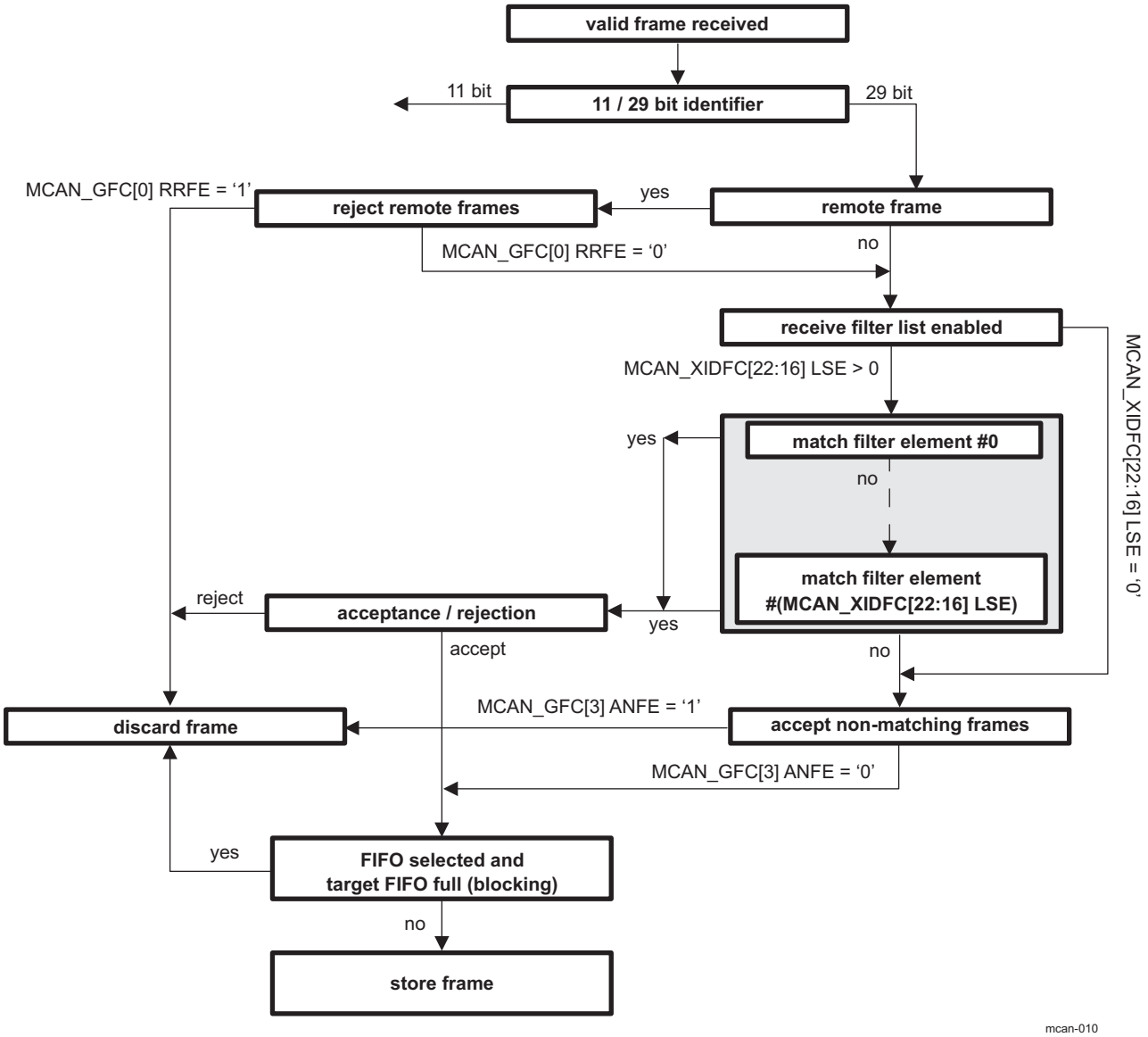
The extended Message ID (29-bit ID) filtering flow is shown in Figure 22-8. Section 22.1.2.11.6, *Extended Message ID Filter Element* describes the extended Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register

Note that before the filter list is executed the received identifier is ANDed with the Extended ID AND Mask (MCAN_XIDAM).

Figure 22-8. Extended Message ID Filter Path



22.1.2.8.2 Rx FIFOs

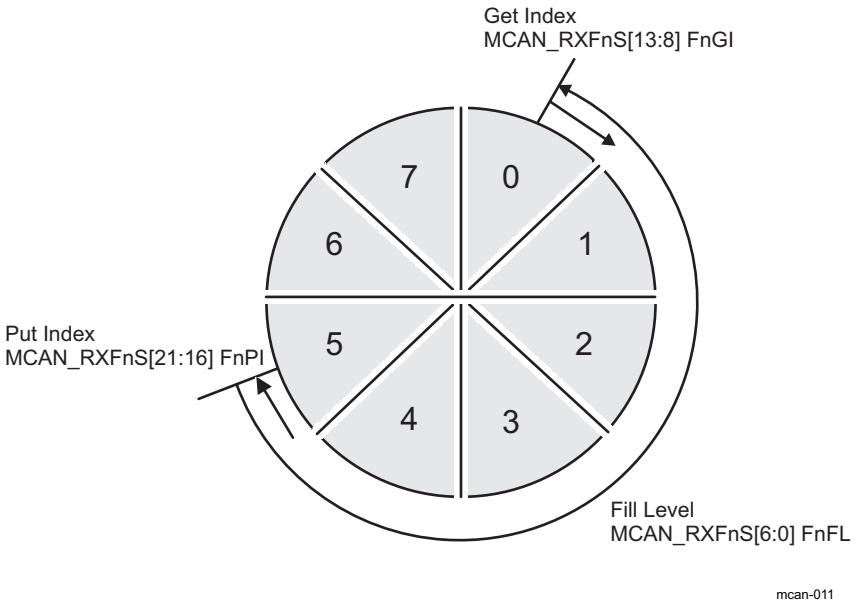
The configuration of the Rx FIFOs (Rx FIFO 0 and Rx FIFO 1) can be done via the MCAN_RXF0C and MCAN_RXF1C registers. Each Rx FIFO can be configured to store up to 64 received messages.

After acceptance filtering the received messages that passed are transferred to the Rx FIFO. The filter mechanisms available for the Rx FIFO 0 and Rx FIFO 1 is described in Section 22.1.2.8.1, *Acceptance Filtering*. Section 22.1.2.11.2, *Rx Buffer and FIFO Element* describes the Rx FIFO element.

The Rx FIFO watermark can be used to prevent an Rx FIFO overflow. If the Rx FIFO fill level reaches the Rx FIFO watermark configured by the MCAN_RXFnC[30:24] FnWM filed (where: n = 0 or 1) an interrupt flag MCAN_IR[1] RF0W / MCAN_IR[5] RF1W is set.

When the Rx FIFO Put Index reaches the Rx FIFO Get Index (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) an Rx FIFO Full condition is signalled by the MCAN_RXFnS[24] FnF status bit and interrupt flag MCAN_IR[2] RF0F / MCAN_IR[6] RF1F is set. Figure 22-9 shows Rx FIFO Status. The FIFOs fill level is presented in the MCAN_RXFnS[6:0] FnFL field (the number of elements stored in Rx FIFO).

Figure 22-9. Rx FIFO Status



Rx FIFOs start address in the Message RAM (MCAN_RXFnC[15:2]FnSA field) have to be configured when reading from an Rx FIFO (Rx FIFO Get Index - MCAN_RXFnS[13:8] FnGI). Table 22-2 presents Rx Buffer / Rx FIFO Element Size for different Rx Buffer / Rx FIFO Data Field Size which is configured via the MCAN_RXESC register.

Table 22-2. Rx Buffer / Rx FIFO Element Size

MCAN_RXESC[10:8] RBDS MCAN_RXESC[2:0] F0DS / MCAN_RXESC[6:4] F1DS	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

22.1.2.8.2.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is the default operation mode for the Rx FIFOs. It is configured by the MCAN_RXFnC[31] FnOM = '0'.

If an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by the MCAN_RXFnS[24] FnF = '1' and interrupt flag MCAN_IR[2] RF0F / MCAN_IR[6] RF1F is set.

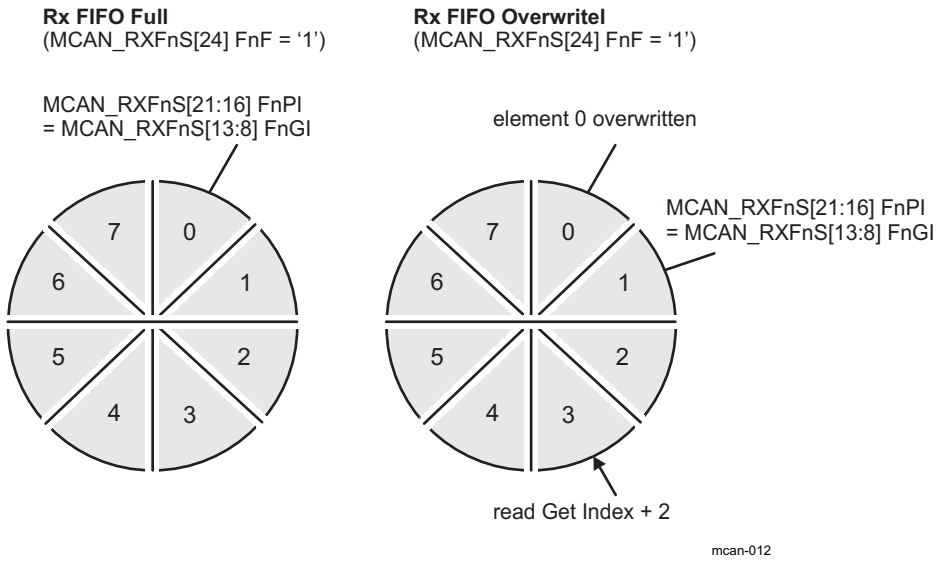
In case a message is received while the corresponding Rx FIFO is full, this message is rejected and the message lost condition is signalled by MCAN_RXFnS[25] RFnL = '1' and interrupt flag MCAN_IR[3] RFnL / MCAN_IR[25] RFnL is set.

22.1.2.8.2.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by the MCAN_RXFnC[31] FnOM = '1'. When an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) signalled by MCAN_RXFnS[24] FnF = '1', the next accepted message for the FIFO will overwrite the oldest FIFO message. Put index / Get index are both incremented by one.

In overwrite mode if an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (Put index) while the Host CPU is reading from the Message RAM (Get index). In this case inconsistent data may be read from the respective Rx FIFO element. The problem is solved by adding an offset to the Get index when reading from the Rx FIFO. The offset depends on how fast the Host CPU accesses the Rx FIFO. Figure 22-10 shows an offset of two with respect to the Get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 22-10. Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA[5:0] FnAI. This increments the get index to that element number. In case the Put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS[24] FnF = '0').

22.1.2.8.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx Buffers. The start address of the Rx Buffers section in the Message RAM is configured via MCAN_RXBC[15:2] RBSA field. To store in an Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = '111' and SFID2 / EFID2[10:9] = '00' has to be configured (see Section 22.1.2.11.5, *Standard Message ID Filter Element* and Section 22.1.2.11.6, *Extended Message ID Filter Element*).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element (the format is the same as for an Rx FIFO element). In addition the flag MCAN_IR[19] DRX (Message stored in Dedicated Rx Buffer) is set.

Table 22-3 shows Example Filter Configuration for Rx Buffers.

Table 22-3. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register MCAN_NDAT1 / MCAN_NDAT2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the Host CPU by writing a '1' to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

22.1.2.8.3.1 Rx Buffer Handling

Rx Buffer Handling include the following steps:

- Reset interrupt flag MCAN_IR[19] DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

22.1.2.9 Tx Handling

The Tx Handler is used to handle the Tx requests. It controls the transfer of transmit messages from the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue to the CAN Core, the Tx Event FIFO, and the Put and Get Index operations. The MCAN module supports up to 32 Tx Buffers. These Tx Buffers can be configured as dedicated Tx Buffers, Tx FIFO, or Tx Queue and as combination of dedicated Tx Buffers / Tx FIFO or dedicated Tx Buffers / Tx Queue. For each Tx Buffer element Classical CAN or CAN FD transmission mode can be configured. Section 22.1.2.11.3 describes the Tx Buffer Element. Table 22-4 shows the possible configurations for message transmission.

Table 22-4. Possible Configurations for Message Transmission

MCAN_CCCR		Tx Buffer Element		Frame Transmission
MCAN_CCCR[9] BRSE	MCAN_CCCR[8] FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	CAN FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	CAN FD without bit rate switching
1	1	1	1	CAN FD with bit rate switching

When the Tx Buffer Request Pending MCAN_TXBRP register is updated, or when a transmission has been started the Tx Handler starts scanning to check for the highest priority pending Tx request. The Tx Buffer with lowest Message ID has highest priority.

NOTE: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

22.1.2.9.1 Transmit Pause

The transmit pause feature is intended for use in CAN networks where the CAN Message IDs are specific and cannot easily be changed. These Message IDs may have a higher priority than other defined Message IDs, while in a specific application their relative priority should be inverse. This allows for a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed (paused).

The transmit pause feature is enabled by the MCAN_CCCR[14] TXP bit. By default this bit is disabled (MCAN_CCCR[14] TXP = '0'). Each time after successfully transmitted message, a pause for two CAN bit times occurs before the start of the next transmission. This allows the other CAN nodes in the network to transmit messages even if their Message IDs have lower priority.

22.1.2.9.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the Host CPU. There are two options:

- Each dedicated Tx Buffer is configured with a specific Message ID.
- Two or more dedicated Tx Buffers are configured with the same Message ID. In this case the Tx Buffer with the lowest buffer number is transmitted first.

After the data section has been updated, a transmission is requested by an Add Request. This is done via the MCAN_TXBAR[x]ARn bit (where x = 0 - 31). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

Table 22-5 shows Tx Buffer / Tx FIFO / Tx Queue Element Size. A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM. The start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index from 0 to 31 (MCAN_TXFQS[20:16] TFQPI) * Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

Table 22-5. Tx Buffer / Tx FIFO / Tx Queue Element Size

MCAN_TXESC[2:0] TBDS	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

22.1.2.9.3 Tx FIFO

Tx FIFO mode is configured by setting bit MCAN_TXBC[30] TFQM = '0'. The stored in the Tx FIFO messages are transmitted starting with the message referenced by the Get Index MCAN_TXFQS[12:8] TFGI field. After each transmission the Get Index is incremented until the Tx FIFO is empty. The Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field indicates the number of the available free Tx FIFO elements. The Tx FIFO allows transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS[20:16] TFQPI field. After each Add Request (MCAN_TXBAR[x] ARn = '1') the Put Index is incemented to the next free Tx FIFO element. When the Put Index reaches the Get Index (MCAN_TXFQS[20:16] TFQPI = MCAN_TXFQS[12:8] TFGI), Tx FIFO Full condition is signalled by bit MCAN_TXFQS[21] TFQF = '1'. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field.

In case a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field is recalculated. In case transmission cancellation is applied to any other Tx Buffer - the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 22-5). The start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS[20:16] TFQPI (from 0 to 31) * Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

22.1.2.9.4 Tx Queue

Tx Queue mode is configured by setting bit MCAN_TXBC[30] TFQM = '1'. The stored in the Tx Queue messages are transmitted starting with the highest priority message (lowest Message ID). In case two or more Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS[20:16] TFQPI field. Each Add Request cyclically increments the Put Index to the next free Tx Buffer. In case of Tx Queue Full condition (MCAN_TXFQS[21] TFQF = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use the MCAN_TXBRP register instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 22-5). The start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS[20:16] TFQPI (from 0 to 31) * Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

22.1.2.9.5 Mixed Dedicated Tx Buffers / Tx FIFO

For this combination the Tx Buffers section in the Message RAM is separated in two parts:

- Dedicated Tx Buffers: the number of Dedicated Tx Buffers is configured by the MCAN_TXBC[21:16] NDTB field
- Tx FIFO: the number of Tx Buffers assigned to the Tx FIFO is configured by the MCAN_TXBC[29:24] TFQS field

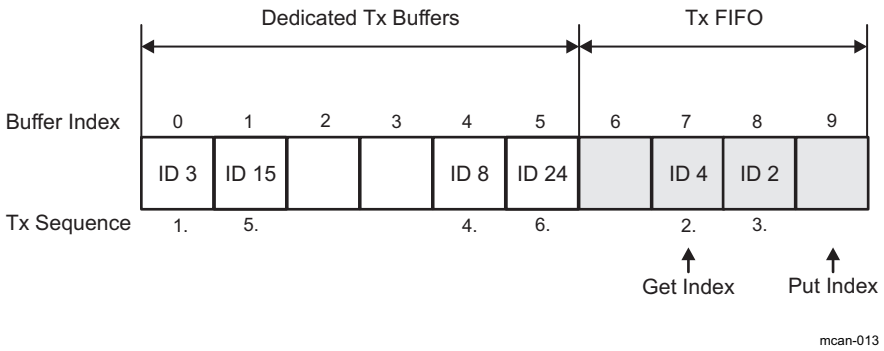
If the MCAN_TXBC[29:24] TFQS field is empty (zero) - only Dedicated Tx Buffers are used.

Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by the MCAN_TXFQS[12:8] TFGI field)
- Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 22-11 shows Mixed Dedicated Tx Buffers / Tx FIFO example.

Figure 22-11. Mixed Dedicated Tx Buffers / Tx FIFO (example)



22.1.2.9.6 Mixed Dedicated Tx Buffers / Tx Queue

For this combination the Tx Buffers section in the Message RAM is separated in two parts:

- Dedicated Tx Buffers: the number of Dedicated Tx Buffers is configured by the MCAN_TXBC[21:16] NDTB field
- Tx Queue: the number of Tx Buffers assigned to the Tx Queue is configured by the MCAN_TXBC[29:24] TFQS field

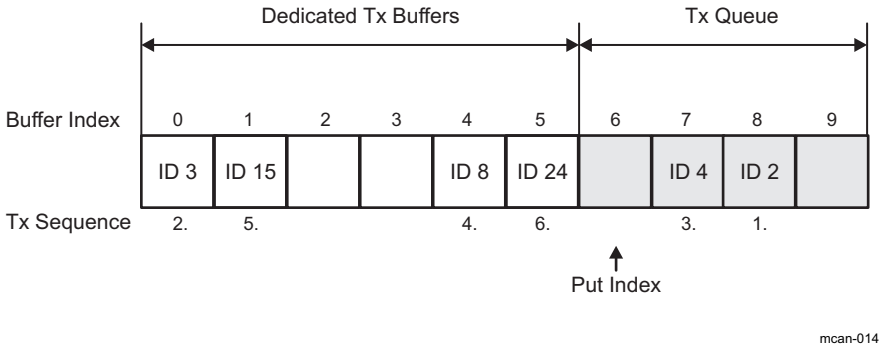
If MCAN_TXBC[29:24] TFQS field is empty (zero) - only Dedicated Tx Buffers are used.

Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 22-12 shows Mixed Dedicated Tx Buffers / Tx Queue example.

Figure 22-12. Mixed Dedicated Tx Buffers / Tx Queue (example)



22.1.2.9.7 Transmit Cancellation

This feature is especially intended for gateway and AUTOSAR based applications. The Host CPU can cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer by setting bit MCAN_TXBCR[n] CRn = '1' (where n = 0 - 31). The corresponding bit position n is equivalent to the number of the Tx Buffer.

Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of the MCAN_TXBCF register (MCAN_TXBCF[n] CFn = '1').

If transmission from a Tx Buffer is already ongoing and a transmit cancellation is requested, the corresponding MCAN_TXBRP[n] TRPn bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO[n] TOn and MCAN_TXBCF[n] CFn bits are set. If the transmission was not successful, only the corresponding bit MCAN_TXBCF[n] CFn = '1'.

NOTE: If pending transmission is cancelled immediately before this transmission could have been started, a short time window occurs where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

22.1.2.9.8 Tx Event Handling

To support Tx Event Handling the Message RAM has implemented a Tx Event FIFO section. Up to 32 Tx Event FIFO elements can be configured. Section 22.1.2.11.4 describes the Tx Event FIFO element. After message transmission on the CAN bus, Message ID and Timestamp are stored in a Tx Event FIFO element. To link a Tx Event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

A Tx Event FIFO full condition is signalled by the MCAN_IR[14] TEFF bit. In this case no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented (MCAN_TXEFS[12:8] EFGI). In case a Tx Event occurs while the Tx Event FIFO is full, this event is rejected and interrupt flag MCAN_IR[15] TEFL bit is set.

The Tx Event FIFO watermark can be configured to avoid a Tx Event FIFO overflow. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by the MCAN_TXEFC[29:24] EFWM field, interrupt flag MCAN_IR[13] TEFW is set. When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS[12:8] EFGI field has to be added to the Tx Event FIFO start address MCAN_TXEFC[15:2] EFSA field.

22.1.2.10 FIFO Acknowledge Handling

The Get Indices of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1) and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see MCAN_RXF0A, MCAN_RXF1A, and MCAN_TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level.

There are two use cases:

- A single element has been read from the FIFO: the Get Index value is written to the FIFO Acknowledge Index.
- A sequence of elements has been read from the FIFO: the Get Index value (Index of the last element read) is written to the FIFO Acknowledge Index at the end of that read sequence.

The Host CPU has free access to the Message RAM. The special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This can be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also changes the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

NOTE: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN module does not check for erroneous values.

22.1.2.11 Message RAM

The MCAN module has implemented Message RAM. The main purpose of the Message RAM is to store:

- Receive Messages
- Transmit Messages
- Tx Event Elements
- Message ID Filter Elements

22.1.2.11.1 Message RAM Configuration

The MCAN module is configured to allocate 4352 words in the Message RAM. The Message RAM has a width of 32 bits.

The address range of the Message RAM is from 0xFF50 0000 to 0xFF50 43FC.

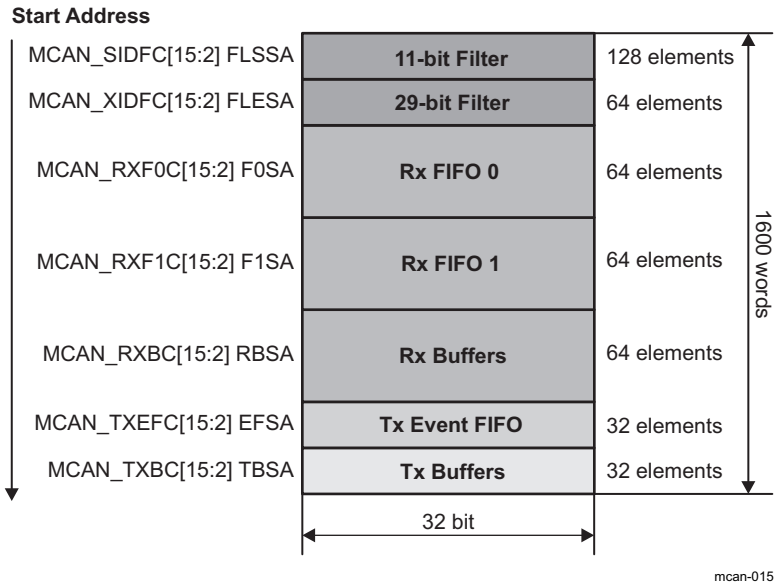
The Message RAM is capable to include each of the sections listed in Figure 22-13. It is not necessary to configure each of the sections (a section in the Message RAM may be 0) and there is not restriction with respect to the sequence of the sections. For parity checking or ECC a respective number of bits has to be added to each word.

When the MCAN module addresses the Message RAM it addresses 32-bit words. The start addresses are configurable and they are 32-bit word addresses.

The element size can be configured for:

- Rx FIFO 0 via the MCAN_RXESC[2:0] F0DS field
- Rx FIFO 1 via the MCAN_RXESC[6:4] F1DS field
- Rx Buffers via the MCAN_RXESC[10:8] RBDS field
- Tx Buffers via the MCAN_TXESC[2:0] TBDS field

Figure 22-13. Message RAM Configuration



The Host CPU configures the following information in the Message RAM:

- Start addresses of the memory sections
- Number of elements in each section
- The size of the elements in some sections

NOTE: The MCAN module does not check for errors in the Message RAM configuration. The configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully. This will prevent falsification or loss of data.

22.1.2.11.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via the MCAN_RXESC register.

Figure 22-14 shows Rx Buffer / Rx FIFO element structure.

Figure 22-14. Rx Buffer / Rx FIFO Element Structure

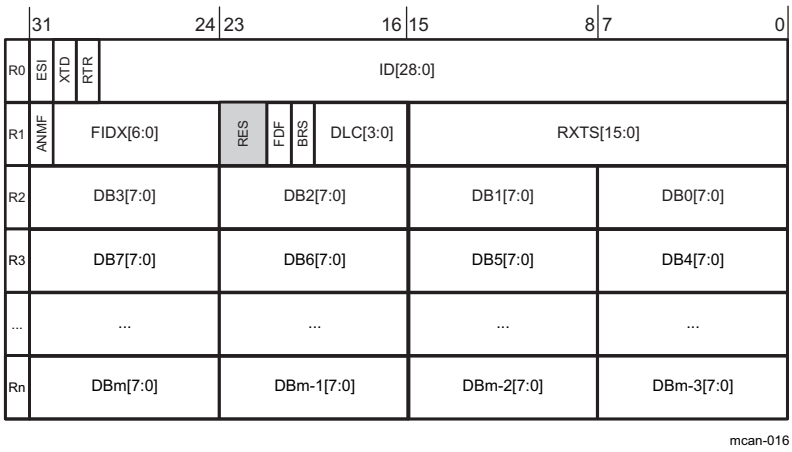


Table 22-6 shows Rx Buffer / Rx FIFO element field descriptions.

Table 22-6. Rx Buffer / Rx FIFO Element Field Descriptions

Word	Bits	Field Name	Description
R0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier Signals to the Host CPU whether the received frame has a standard or extended identifier. • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request Signals to the Host CPU whether the received frame is a data frame or a remote frame. • 0x0: Received frame is a data frame • 0x1: Received frame is a remote frame **Note:** There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = '1'), RTR bit reflects the state of the reserved r1 bit (RES[23]).
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier is stored into ID[28:18].
R1	31	ANMF	Accepted Non-matching Frame Acceptance of non-matching frames may be enabled via the MCAN_GFC[5:4] ANFS and MCAN_GFC[3:2] ANFE fields. • 0x0: Received frame matching filter index FIDX field • 0x1: Received frame did not match any Rx filter element
	30:24	FIDX[6:0]	Filter Index 0x0-0x7F (0-127): Index of matching Rx acceptance filter element (invalid if ANMF = '1'). Range is 0 to MCAN_SIDFC[23:16] LSS - 1 respectively MCAN_XIDFC[22:16] LSE - 1.
	23:22	RES	Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame received without bit rate switching • 0x1: Frame received with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: received frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: received frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: received frame has 12/16/20/24/32/48/64 data bytes
	15:0	RXTS[15:0]	Rx Timestamp Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC[19:16] TCP.
R2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0

Table 22-6. Rx Buffer / Rx FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
R3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Rn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3-17) are used for storage of a CAN message's data field.

22.1.2.11.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler makes difference between dedicated Tx Buffers and Tx FIFO / Tx Queue via the MCAN_TXBC[29:24] TFQS and MCAN_TXBC[21:16] NDTB fields. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via the MCAN_TXESC register.

Figure 22-15 shows Tx Buffer element structure.

Figure 22-15. Tx Buffer Element Structure

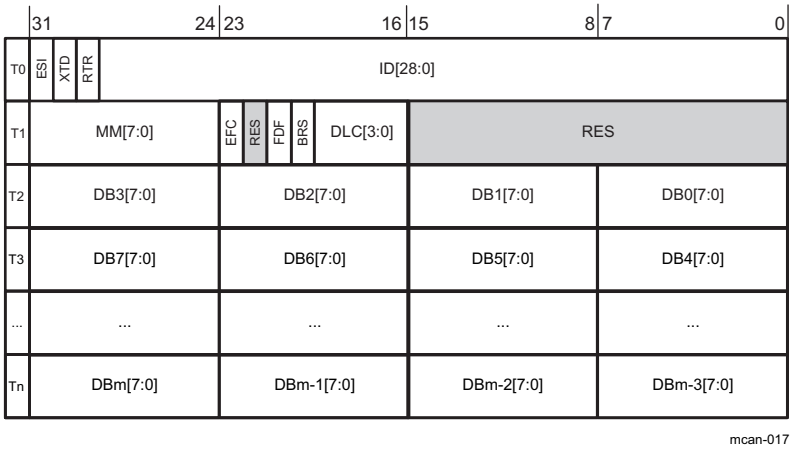


Table 22-7 shows Tx Buffer element field descriptions.

Table 22-7. Tx Buffer Element Field Descriptions

Word	Bits	Field Name	Description
T0	31	ESI	Error State Indicator • 0x0: ESI bit in CAN FD format depends only on error passive flag • 0x1: ESI bit in CAN FD format transmitted recessive **Note:** The ESI bit of the transmit buffer is or'ed with the error passive flag to decide the value of the ESI bit in the transmitted CAN FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Transmit data frame • 0x1: Transmit remote frame **Note:** When RTR = 1, the MCAN module transmits a remote frame according to ISO11898-1, even if the MCAN_CCCR[8] FDOE bit enables the transmission in CAN FD format.
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].
T1	31:24	MM[7:0]	Message Marker Written by Host CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 22-8).
	23	EFC	Event FIFO Control • 0x0: Don't store Tx events • 0x1: Store Tx events
	22	RES	Reserved
	21	FDF	FD Format • 0x0: Frame transmitted in Classic CAN format • 0x1: Frame transmitted in CAN FD format
	20	BRS	Bit Rate Switch • 0x0: CAN FD frames transmitted without bit rate switching • 0x1: CAN FD frames transmitted with bit rate switching **Note:** ESI, FDF, and BRS bits are only evaluated when CAN FD operation is enabled vie the MCAN_CCCR[8] FDOE bit. BRS bit is only evaluated when in addition the MCAN_CCCR[9] BRSE = '1'.
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: transmit frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: transmit frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes
	15:0	RES	Reserved
T2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0

Table 22-7. Tx Buffer Element Field Descriptions (continued)

Word	Bits	Field Name	Description
T3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Tn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3-17) are used for storage of a CAN message's data field.

22.1.2.11.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from the MCAN_TXEFS register.

Figure 22-16 shows Tx Event FIFO element structure.

Figure 22-16. Tx Event FIFO Element Structure

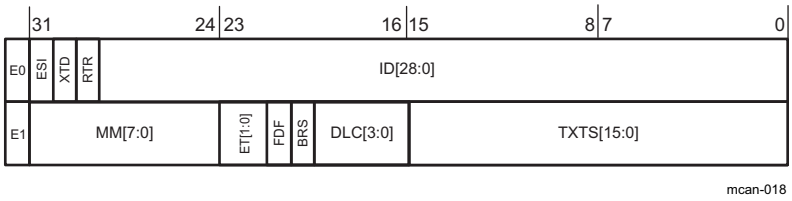


Table 22-8 shows Tx Event FIFO element field descriptions.

Table 22-8. Tx Event FIFO Element Field Descriptions

Word	Bits	Field Name	Description
E0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Data frame transmitted • 0x1: Remote frame transmitted
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

Table 22-8. Tx Event FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
E1	31:24	MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 22-7).
	23:22	ET[1:0]	Event Type • 0x0: Reserved • 0x1: Tx event • 0x2: Transmission in spite of cancellation (always set for transmissions in DAR mode) • 0x3: Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame transmitted without bit rate switching • 0x1: Frame transmitted with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: frame with 0-8 data bytes transmitted • 0x9-0xF (9-15): CAN: frame with 8 data bytes transmitted • 0x9-0xF (9-15): CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted
	15:0	TXTS[15:0]	Tx Timestamp Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC[19:16] TCP filed.

22.1.2.11.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address MCAN_SIDFC[15:2] FLSSA field plus the index of the filter element (0-127).

Figure 22-17 shows Standard Message ID Filter element structure.

Figure 22-17. Standard Message ID Filter Element Structure

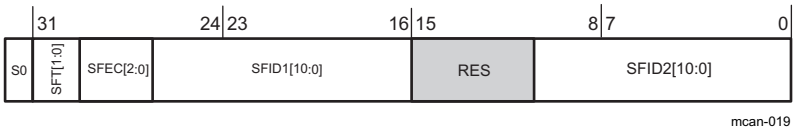


Table 22-9 shows Standard Message ID Filter element field descriptions.

Table 22-9. Standard Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
S0	31:30	SFT[1:0]	Standard Filter Type • 0x0: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) • 0x1: Dual ID filter for SFID1 or SFID2 • 0x2: Classic filter: SFID1 = filter; SFID2 = mask • 0x3: Filter element disabled **Note:** With SFT = '11' the filter element is disabled and the acceptance filtering continues (same behaviour as with SFEC = '000')
	29:27	SFEC[2:0]	Standard Filter Element Configuration All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = '100', '101', or '110' a match sets interrupt flag MCAN_IR[8]HPM and, if enabled, an interrupt is generated. In this case the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer, configuration of SFT[1:0] ignored
	26:16	SFID1[10:0]	Standard Filter ID 1 When filtering for Rx Buffers this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.
	15:11	RES	Reserved
	10:0	SFID2[10:0]	Standard Filter ID 2 This bit field has a different meaning depending on the configuration of SFEC: • 1) SFEC = '001' - '110' Second ID of standard ID filter element • 2) SFEC = '111' Filter for Rx Buffers
		SFID2[10:9]	This filed is decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		SFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		SFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC[15:2] RBSA field for storage of a matching message.

22.1.2.11.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC[15:2] FLESA field plus two times the index of the filter element (0-63).

Figure 22-18 shows Extended Message ID Filter element structure.

Figure 22-18. Extended Message ID Filter Element Structure

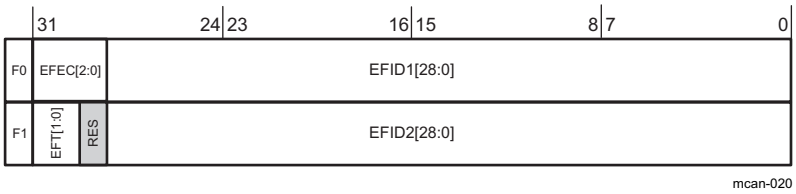


Table 22-10 shows Extended Message ID Filter element field descriptions.

Table 22-10. Extended Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
F0	31:29	EFEC[2:0]	Extended Filter Element Configuration All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = '100', '101', or '110' a match sets interrupt flag MCAN_IR[8]HPM and, if enabled, an interrupt is generated. In this case the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored
	28:0	EFID1[28:0]	Extended Filter ID 1 First ID of extended ID filter element. When filtering for Rx Buffers this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see Section 22.1.2.8.1.5, *Extended Message ID Filtering*) is used.

Table 22-10. Extended Message ID Filter Element Field Descriptions (continued)

Word	Bits	Field Name	Description
F1	31:30	EFT[1:0]	Extended Filter Type • 0x0: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) • 0x1: Dual ID filter for EFID1 or EFID2 • 0x2: Classic filter: EFID1 = filter, EFID2 = mask • 0x3: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), XIDAM mask not applied
	29	RES	Reserved
	28:0	EFID2[28:0]	Extended Filter ID 2 This bit field has a different meaning depending on the configuration of EFEC: • 1) EFEC = '001' - '110' Second ID of extended ID filter element • 2) EFEC = '111' Filter for Rx Buffers
		EFID2[10:9]	This field decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		EFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		EFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC[15:2] RBSA field for storage of a matching message.

22.1.3 MCAN Register Manual

22.1.3.1 MSS_MCAN_CFG Registers

Table 22-11 lists the memory-mapped registers for the MSS_MCAN_CFG. All register offset addresses not listed in Table 22-11 should be considered as reserved locations and the register contents should not be modified.

Table 22-11. MSS_MCAN_CFG Registers

Offset	Acronym	Register Name	Section
0h	MCANSS_PID	MCANSS_PID	Section 22.1.3.1.1
4h	MCANSS_CTRL	MCANSS_CTRL	Section 22.1.3.1.2
8h	MCANSS_STAT	MCANSS_STAT	Section 22.1.3.1.3
Ch	MCANSS_ICS	MCANSS_ICS	Section 22.1.3.1.4
10h	MCANSS_IRS	MCANSS_IRS	Section 22.1.3.1.5
14h	MCANSS_IECS	MCANSS_IECS	Section 22.1.3.1.6
18h	MCANSS_IE	MCANSS_IE	Section 22.1.3.1.7
1Ch	MCANSS_IES	MCANSS_IES	Section 22.1.3.1.8
20h	MCANSS_EOI	MCANSS_EOI	Section 22.1.3.1.9
24h	MCANSS_EXT_TS_PS	MCANSS_EXT_TS_PS	Section 22.1.3.1.10
28h	MCANSS_EXT_TS_USIC	MCANSS_EXT_TS_USIC	Section 22.1.3.1.11
80h	MCANSS_ECC_EOI	MCANSS_ECC_EOI	Section 22.1.3.1.12
200h	MCAN_CREL	MCAN_CREL	Section 22.1.3.1.13
204h	MCAN_ENDN	MCAN_ENDN	Section 22.1.3.1.14
208h	MCAN_CUST	MCAN_CUST	Section 22.1.3.1.15
20Ch	MCAN_DBTP	MCAN_DBTP	Section 22.1.3.1.16
210h	MCAN_TEST	MCAN_TEST	Section 22.1.3.1.17
214h	MCAN_RWD	MCAN_RWD	Section 22.1.3.1.18
218h	MCAN_CCCR	MCAN_CCCR	Section 22.1.3.1.19
21Ch	MCAN_NBTP	MCAN_NBTP	Section 22.1.3.1.20
220h	MCAN_TSCC	MCAN_TSCC	Section 22.1.3.1.21
224h	MCAN_TSCV	MCAN_TSCV	Section 22.1.3.1.22
228h	MCAN_TOCC	MCAN_TOCC	Section 22.1.3.1.23
22Ch	MCAN_TOCV	MCAN_TOCV	Section 22.1.3.1.24
230h	MCAN_RES00	MCAN_RES00	Section 22.1.3.1.25
234h	MCAN_RES01	MCAN_RES01	Section 22.1.3.1.26
238h	MCAN_RES02	MCAN_RES02	Section 22.1.3.1.27
23Ch	MCAN_RES03	MCAN_RES03	Section 22.1.3.1.28
240h	MCAN_ECR	MCAN_ECR	Section 22.1.3.1.29
244h	MCAN_PSR	MCAN_PSR	Section 22.1.3.1.30
248h	MCAN_TDCR	MCAN_TDCR	Section 22.1.3.1.31
24Ch	MCAN_RES04	MCAN_RES04	Section 22.1.3.1.32
250h	MCAN_IR	MCAN_IR	Section 22.1.3.1.33
254h	MCAN_IE	MCAN_IE	Section 22.1.3.1.34
258h	MCAN_ILS	MCAN_ILS	Section 22.1.3.1.35
25Ch	MCAN_ILE	MCAN_ILE	Section 22.1.3.1.36
260h	MCAN_RES05	MCAN_RES05	Section 22.1.3.1.37
264h	MCAN_RES06	MCAN_RES06	Section 22.1.3.1.38
268h	MCAN_RES07	MCAN_RES07	Section 22.1.3.1.39
26Ch	MCAN_RES08	MCAN_RES08	Section 22.1.3.1.40
270h	MCAN_RES09	MCAN_RES09	Section 22.1.3.1.41
274h	MCAN_RES10	MCAN_RES10	Section 22.1.3.1.42
278h	MCAN_RES11	MCAN_RES11	Section 22.1.3.1.43

Table 22-11. MSS_MCAN_CFG Registers (continued)

Offset	Acronym	Register Name	Section
27Ch	MCAN_RES12	MCAN_RES12	Section 22.1.3.1.44
280h	MCAN_GFC	MCAN_GFC	Section 22.1.3.1.45
284h	MCAN_SIDFC	MCAN_SIDFC	Section 22.1.3.1.46
288h	MCAN_XIDFC	MCAN_XIDFC	Section 22.1.3.1.47
28Ch	MCAN_RES13	MCAN_RES13	Section 22.1.3.1.48
290h	MCAN_XIDAM	MCAN_XIDAM	Section 22.1.3.1.49
294h	MCAN_HPMS	MCAN_HPMS	Section 22.1.3.1.50
298h	MCAN_NDAT1	MCAN_NDAT1	Section 22.1.3.1.51
29Ch	MCAN_NDAT2	MCAN_NDAT2	Section 22.1.3.1.52
2A0h	MCAN_RXF0C	MCAN_RXF0C	Section 22.1.3.1.53
2A4h	MCAN_RXF0S	MCAN_RXF0S	Section 22.1.3.1.54
2A8h	MCAN_RXF0A	MCAN_RXF0A	Section 22.1.3.1.55
2ACh	MCAN_RXBC	MCAN_RXBC	Section 22.1.3.1.56
2B0h	MCAN_RXF1C	MCAN_RXF1C	Section 22.1.3.1.57
2B4h	MCAN_RXF1S	MCAN_RXF1S	Section 22.1.3.1.58
2B8h	MCAN_RXF1A	MCAN_RXF1A	Section 22.1.3.1.59
2BCh	MCAN_RXESC	MCAN_RXESC	Section 22.1.3.1.60
2C0h	MCAN_TXBC	MCAN_TXBC	Section 22.1.3.1.61
2C4h	MCAN_TXFQS	MCAN_TXFQS	Section 22.1.3.1.62
2C8h	MCAN_TXESC	MCAN_TXESC	Section 22.1.3.1.63
2CCh	MCAN_TXBRP	MCAN_TXBRP	Section 22.1.3.1.64
2D0h	MCAN_TXBAR	MCAN_TXBAR	Section 22.1.3.1.65
2D4h	MCAN_TXBCR	MCAN_TXBCR	Section 22.1.3.1.66
2D8h	MCAN_TXBTO	MCAN_TXBTO	Section 22.1.3.1.67
2DCh	MCAN_TXBCF	MCAN_TXBCF	Section 22.1.3.1.68
2E0h	MCAN_TXBTIE	MCAN_TXBTIE	Section 22.1.3.1.69
2E4h	MCAN_TXBCIE	MCAN_TXBCIE	Section 22.1.3.1.70
2E8h	MCAN_RES14	MCAN_RES14	Section 22.1.3.1.71
2ECh	MCAN_RES15	MCAN_RES15	Section 22.1.3.1.72
2F0h	MCAN_TXEFC	MCAN_TXEFC	Section 22.1.3.1.73
2F4h	MCAN_TXEFS	MCAN_TXEFS	Section 22.1.3.1.74
2F8h	MCAN_TXEFA	MCAN_TXEFA	Section 22.1.3.1.75
2FCh	MCAN_RES16	MCAN_RES16	Section 22.1.3.1.76

Complex bit access types are encoded to fit into small table cells. Table 22-12 shows the codes that are used for access types in this section.

Table 22-12. MSS_MCAN_CFG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W W	Write Write
W0C	W 0C	Write 0 to clear

22.1.3.1.1 MCANSS_PID Register (Offset = 0h) [reset = 68E02A00h]

MCANSS_PID is shown in Figure 22-19 and described in Table 22-13.

Return to Summary Table.

SS_PID

Figure 22-19. MCANSS_PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		BU		MODULE_ID											
R-1h		R-2h		R-8E0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-5h					R-2h			R-0h		R-0h					

Table 22-13. MCANSS_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID register scheme
29-28	BU	R	2h	Business Unit: 10 = Processors
27-16	MODULE_ID	R	8E0h	Module ID
15-11	RTL	R	5h	RTL revision. Will vary depending on release.
10-8	MAJOR	R	2h	Major revision
7-6	CUSTOM	R	0h	Custom
5-0	MINOR	R	0h	Minor revision

22.1.3.1.2 MCANSS_CTRL Register (Offset = 4h) [reset = 0h]

MCANSS_CTRL is shown in Figure 22-20 and described in Table 22-14.

Return to Summary Table.

SS_CTRL

Figure 22-20. MCANSS_CTRL Register

31	30	29	28	27	26	25	24
NU0							
R-0h							
23	22	21	20	19	18	17	16
NU0							
R-0h							
15	14	13	12	11	10	9	8
NU0							
R-0h							
7	6	5	4	3	2	1	0
NU0	EXT_TS_CNTR_EN	AUTOWAKEUP	WAKEUPREGEN	EMUEN	EMUFACK	CLKFACK	RESET
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W0C-0h

Table 22-14. MCANSS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	NU0	R	0h	Reserved
6	EXT_TS_CNTR_EN	R/W	0h	External TimeStamp Counter Enable
5	AUTOWAKEUP	R/W	0h	Automatic Wakeup Enable
4	WAKEUPREGEN	R/W	0h	Wakeup Request Enable
3	EMUEN	R/W	0h	Emulation Enable
2	EMUFACK	R/W	0h	Emulation Fast Ack
1	CLKFACK	R/W	0h	Clock Fast Ack
0	RESET	R/W0C	0h	Initiates a Soft Reset

22.1.3.1.3 MCANSS_STAT Register (Offset = 8h) [reset = 0h]

MCANSS_STAT is shown in Figure 22-21 and described in Table 22-15.

Return to Summary Table.

SS_STAT

Figure 22-21. MCANSS_STAT Register

31	30	29	28	27	26	25	24
NU1							
R-0h							
23	22	21	20	19	18	17	16
NU1							
R-0h							
15	14	13	12	11	10	9	8
NU1							
R-0h							
7	6	5	4	3	2	1	0
NU1					EN_FDOE	MMI_DONE	RESET_STS
R-0h					R-0h	R-0h	R-0h

Table 22-15. MCANSS_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU1	R	0h	Reserved
2	EN_FDOE	R	0h	Enable_FDOE configuration
1	MMI_DONE	R	0h	0:Memory Initialization is in progress, 1:Memory Intialization Done
0	RESET_STS	R	0h	1:Reset is in progress, 0:Not in reset

22.1.3.1.4 MCANSS_ICS Register (Offset = Ch) [reset = 0h]

MCANSS_ICS is shown in Figure 22-22 and described in Table 22-16.

Return to Summary Table.

SS_ICS

Figure 22-22. MCANSS_ICS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2															ICS
R-0h															R/W0 C-0h

Table 22-16. MCANSS_ICS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU2	R	0h	Reserved
0	ICS	R/W0C	0h	External TimeStamp Counter Overflow Interrupt status. Write '1' to clear bits.

22.1.3.1.5 MCANSS_IRS Register (Offset = 10h) [reset = 0h]

MCANSS_IRS is shown in Figure 22-23 and described in Table 22-17.

Return to Summary Table.

SS_IRS

Figure 22-23. MCANSS_IRS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3															IRS
R-0h															R-0h

Table 22-17. MCANSS_IRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU3	R	0h	Reserved
0	IRS	R	0h	External TimeStamp Counter Overflow Interrupt status.

22.1.3.1.6 MCANSS_IECS Register (Offset = 14h) [reset = 0h]

MCANSS_IECS is shown in Figure 22-24 and described in Table 22-18.

Return to Summary Table.

SS_IECS

Figure 22-24. MCANSS_IECS Register

31	30	29	28	27	26	25	24
NU4							
R-0h							
23	22	21	20	19	18	17	16
NU4							
R-0h							
15	14	13	12	11	10	9	8
NU4							
R-0h							
7	6	5	4	3	2	1	0
NU4							IECS
R-0h							R/W0C-0h

Table 22-18. MCANSS_IECS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU4	R	0h	Reserved
0	IECS	R/W0C	0h	External TimeStamp Counter Overflow Interrupt. Write '1' to clear bits.

22.1.3.1.7 MCANSS_IE Register (Offset = 18h) [reset = 0h]

MCANSS_IE is shown in Figure 22-25 and described in Table 22-19.

Return to Summary Table.

SS_IE

Figure 22-25. MCANSS_IE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU5															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU5															IE
R-0h															R-0h

Table 22-19. MCANSS_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU5	R	0h	Reserved
0	IE	R	0h	External TimeStamp Counter Overflow Interrupt.

22.1.3.1.8 MCANSS_IES Register (Offset = 1Ch) [reset = 0h]

MCANSS_IES is shown in Figure 22-26 and described in Table 22-20.

Return to Summary Table.

SS_IES

Figure 22-26. MCANSS_IES Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU6															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU6															IES
R-0h															R-0h

Table 22-20. MCANSS_IES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU6	R	0h	Reserved
0	IES	R	0h	External TimeStamp Counter Overflow Interrupt. Read Enabled Interrupts

22.1.3.1.9 MCANSS_EOI Register (Offset = 20h) [reset = 0h]

MCANSS_EOI is shown in Figure 22-27 and described in Table 22-21.

Return to Summary Table.

SS_EOI

Figure 22-27. MCANSS_EOI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU7																								EOI							
R-0h																								R/W-0h							

Table 22-21. MCANSS_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU7	R	0h	Reserved
7-0	EOI	R/W	0h	Pulse output corresponding to interrupt associated with value written to EOI register if interrupt is still asserted

22.1.3.1.10 MCANSS_EXT_TS_PS Register (Offset = 24h) [reset = 0h]

MCANSS_EXT_TS_PS is shown in Figure 22-28 and described in Table 22-22.

Return to Summary Table.

SS_EXT_TS_PS

Figure 22-28. MCANSS_EXT_TS_PS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU8								PRESCALE																							
R-0h								R/W-0h																							

Table 22-22. MCANSS_EXT_TS_PS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU8	R	0h	Reserved
23-0	PRESCALE	R/W	0h	Pulse output corresponding to interrupt associated with value written to EOI register if interrupt is still asserted

22.1.3.1.11 MCANSS_EXT_TS_USIC Register (Offset = 28h) [reset = 0h]

MCANSS_EXT_TS_USIC is shown in Figure 22-29 and described in Table 22-23.

Return to Summary Table.

SS_EXT_TS_USIC

Figure 22-29. MCANSS_EXT_TS_USIC Register

31	30	29	28	27	26	25	24
NU9							
R-0h							
23	22	21	20	19	18	17	16
NU9							
R-0h							
15	14	13	12	11	10	9	8
NU9							
R-0h							
7	6	5	4	3	2	1	0
NU9			EXT_TS_INTR_CNTR				
R-0h			R-0h				

Table 22-23. MCANSS_EXT_TS_USIC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	NU9	R	0h	Reserved
4-0	EXT_TS_INTR_CNTR	R	0h	Number of unserviced rollover interupts. If >0 an EOI will issue another pulse interrupt

22.1.3.1.12 MCANSS_ECC_EOI Register (Offset = 80h) [reset = 0h]

MCANSS_ECC_EOI is shown in Figure 22-30 and described in Table 22-24.

Return to Summary Table.

SS_ECC_EOI

Figure 22-30. MCANSS_ECC_EOI Register

31	30	29	28	27	26	25	24
NU11							
R-0h							
23	22	21	20	19	18	17	16
NU11							
R-0h							
15	14	13	12	11	10	9	8
NU11							ECC_EOI
R-0h							R/W0C-0h
7	6	5	4	3	2	1	0
NU10							
R-0h							

Table 22-24. MCANSS_ECC_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	NU11	R	0h	Reserved
8	ECC_EOI	R/W0C	0h	ECC EOI
7-0	NU10	R	0h	Reserved

22.1.3.1.13 MCAN_CREL Register (Offset = 200h) [reset = 32150320h]

MCAN_CREL is shown in Figure 22-31 and described in Table 22-25.

Return to Summary Table.

CREL

Figure 22-31. MCAN_CREL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL				STEP				SUBSTEP				YEAR			
R-3h				R-2h				R-1h				R-5h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON								DAY							
R-3h								R-20h							

Table 22-25. MCAN_CREL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	REL	R	3h	Core Release
27-24	STEP	R	2h	Step of Core Release
23-20	SUBSTEP	R	1h	Sub-Step of Core Release
19-16	YEAR	R	5h	Time Stamp Year
15-8	MON	R	3h	Time Stamp Month
7-0	DAY	R	20h	Time Stamp Day

22.1.3.1.14 MCAN_ENDN Register (Offset = 204h) [reset = 87654321h]

MCAN_ENDN is shown in Figure 22-32 and described in Table 22-26.

Return to Summary Table.

ENDN

Figure 22-32. MCAN_ENDN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV																															
R-87654321h																															

Table 22-26. MCAN_ENDN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ETV	R	87654321h	Endianess test value

22.1.3.1.15 MCAN_CUST Register (Offset = 208h) [reset = 0h]

MCAN_CUST is shown in Figure 22-33 and described in Table 22-27.

Return to Summary Table.

CUST

Figure 22-33. MCAN_CUST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST																															
R-0h																															

Table 22-27. MCAN_CUST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CUST	R	0h	Custom

22.1.3.1.16 MCAN_DBTP Register (Offset = 20Ch) [reset = A33h]

MCAN_DBTP is shown in Figure 22-34 and described in Table 22-28.

Return to Summary Table.

DBTP

Figure 22-34. MCAN_DBTP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU13								TDC	NU12		DBRP				
R-0h								R/W-0h	R-0h		R/W-0h				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU11			DTSEG1					DTSEG2				DSJW			
R-0h			R/W-Ah					R/W-3h				R/W-3h			

Table 22-28. MCAN_DBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU13	R	0h	Reserved
23	TDC	R/W	0h	Transmitter Delay Compensation
22-21	NU12	R	0h	Reserved
20-16	DBRP	R/W	0h	Data Baud Rate Prescaler
15-13	NU11	R	0h	Reserved
12-8	DTSEG1	R/W	Ah	Data time segment before smaple point
7-4	DTSEG2	R/W	3h	Data time segment after sample point
3-0	DSJW	R/W	3h	Data resynchronization Jump Width

22.1.3.1.17 MCAN_TEST Register (Offset = 210h) [reset = 0h]

MCAN_TEST is shown in Figure 22-35 and described in Table 22-29.

Return to Summary Table.

TEST

Figure 22-35. MCAN_TEST Register

31	30	29	28	27	26	25	24
NU15							
R-0h							
23	22	21	20	19	18	17	16
NU15							
R-0h							
15	14	13	12	11	10	9	8
NU15							
R-0h							
7	6	5	4	3	2	1	0
RX	TX		LBCK	NU14			
R-0h	R/W-0h		R/W-0h	R-0h			

Table 22-29. MCAN_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU15	R	0h	Reserved
7	RX	R	0h	Receive Pin
6-5	TX	R/W	0h	Control of Transmit Pin
4	LBCK	R/W	0h	Loop Back Mode
3-0	NU14	R	0h	Reserved

22.1.3.1.18 MCAN_RWD Register (Offset = 214h) [reset = 0h]

MCAN_RWD is shown in Figure 22-36 and described in Table 22-30.

Return to Summary Table.

RWD

Figure 22-36. MCAN_RWD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU16																WDV								WDC							
R-0h																R-0h								R/W-0h							

Table 22-30. MCAN_RWD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU16	R	0h	Reserved
15-8	WDV	R	0h	Watchdog Value
7-0	WDC	R/W	0h	Watchdog Counter Value

22.1.3.1.19 MCAN_CCCR Register (Offset = 218h) [reset = 0h]

MCAN_CCCR is shown in Figure 22-37 and described in Table 22-31.

Return to Summary Table.

CCCR

Figure 22-37. MCAN_CCCR Register

31	30	29	28	27	26	25	24
NU18							
R/W-0h							
23	22	21	20	19	18	17	16
NU18							
R/W-0h							
15	14	13	12	11	10	9	8
NU18	TXP	EFBI	PXHD	NU17		BRSE	FDOE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-31. MCAN_CCCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	NU18	R/W	0h	Reserved
14	TXP	R/W	0h	Transmit Pause
13	EFBI	R/W	0h	Edge Filtering durign Bus Integration
12	PXHD	R/W	0h	Protocol Exception Handling Disable
11-10	NU17	R	0h	Reserved
9	BRSE	R/W	0h	Bit Rate Switch Enable
8	FDOE	R/W	0h	FD Operation Enable
7	TEST	R/W	0h	Test Mode enable
6	DAR	R/W	0h	Disable Automatic Regransmission
5	MON	R/W	0h	Bus Monitoring Mode
4	CSR	R/W	0h	Clock Stop Request
3	CSA	R	0h	Clock Stop Acknowledge
2	ASM	R/W	0h	Restriced Operation Mode
1	CCE	R/W	0h	Configuration Change Enable
0	INIT	R/W	0h	Initialization

22.1.3.1.20 MCAN_NBTP Register (Offset = 21Ch) [reset = 06000A03h]

MCAN_NBTP is shown in Figure 22-38 and described in Table 22-32.

Return to Summary Table.

NBTP

Figure 22-38. MCAN_NBTP Register

31	30	29	28	27	26	25	24
NSJW							NBRP
R/W-3h							R/W-0h
23	22	21	20	19	18	17	16
NBRP							
R/W-0h							
15	14	13	12	11	10	9	8
NTSEG1							
R/W-Ah							
7	6	5	4	3	2	1	0
NU19	NTSEG2						
R-0h	R/W-3h						

Table 22-32. MCAN_NBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NSJW	R/W	3h	Nominal Resynchronization Jump Width
24-16	NBRP	R/W	0h	Nominal Baud Rate Prescaler
15-8	NTSEG1	R/W	Ah	Nominal Time segment before sample point
7	NU19	R	0h	Reserved
6-0	NTSEG2	R/W	3h	Nominal Time segment after sample point

22.1.3.1.21 MCAN_TSCC Register (Offset = 220h) [reset = 0h]

MCAN_TSCC is shown in Figure 22-39 and described in Table 22-33.

Return to Summary Table.

TSCC

Figure 22-39. MCAN_TSCC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU21												TCP			
R-0h												R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU20														TSS	
R-0h														R/W-0h	

Table 22-33. MCAN_TSCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU21	R	0h	Reserved
19-16	TCP	R/W	0h	Timestamp Counter Prescaler
15-2	NU20	R	0h	Reserved
1-0	TSS	R/W	0h	Timestamp Select

22.1.3.1.22 MCAN_TSCV Register (Offset = 224h) [reset = 0h]

MCAN_TSCV is shown in Figure 22-40 and described in Table 22-34.

Return to Summary Table.

TSCV

Figure 22-40. MCAN_TSCV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU22																TSC															
R-0h																R/W-0h															

Table 22-34. MCAN_TSCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU22	R	0h	Reserved
15-0	TSC	R/W	0h	Timestamp Counter

22.1.3.1.23 MCAN_TOCC Register (Offset = 228h) [reset = FFFF0000h]

MCAN_TOCC is shown in Figure 22-41 and described in Table 22-35.

Return to Summary Table.

TOCC

Figure 22-41. MCAN_TOCC Register

31	30	29	28	27	26	25	24
TOP							
R/W-FFFFh							
23	22	21	20	19	18	17	16
TOP							
R/W-FFFFh							
15	14	13	12	11	10	9	8
NU23							
R-0h							
7	6	5	4	3	2	1	0
NU23					TOS		ETOC
R-0h					R/W-0h		R/W-0h

Table 22-35. MCAN_TOCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TOP	R/W	FFFFh	Timeout Period
15-3	NU23	R	0h	Reserved
2-1	TOS	R/W	0h	Timeout Select
0	ETOC	R/W	0h	Enable Timeout Counter

22.1.3.1.24 MCAN_TOCV Register (Offset = 22Ch) [reset = FFFFh]

MCAN_TOCV is shown in Figure 22-42 and described in Table 22-36.

Return to Summary Table.

TOCV

Figure 22-42. MCAN_TOCV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU24																TOC															
R-0h																R/W-FFFFh															

Table 22-36. MCAN_TOCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU24	R	0h	Reserved
15-0	TOC	R/W	FFFFh	Timeout Counter

22.1.3.1.25 MCAN_RES00 Register (Offset = 230h) [reset = 0h]

MCAN_RES00 is shown in Figure 22-43 and described in Table 22-37.

Return to Summary Table.

RES00

Figure 22-43. MCAN_RES00 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES00																															
R-0h																															

Table 22-37. MCAN_RES00 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES00	R	0h	Reserved

22.1.3.1.26 MCAN_RES01 Register (Offset = 234h) [reset = 0h]

MCAN_RES01 is shown in Figure 22-44 and described in Table 22-38.

Return to Summary Table.

RES01

Figure 22-44. MCAN_RES01 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES01																															
R-0h																															

Table 22-38. MCAN_RES01 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES01	R	0h	Reserved

22.1.3.1.27 MCAN_RES02 Register (Offset = 238h) [reset = 0h]

MCAN_RES02 is shown in Figure 22-45 and described in Table 22-39.

Return to Summary Table.

RES02

Figure 22-45. MCAN_RES02 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES02																															
R-0h																															

Table 22-39. MCAN_RES02 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES02	R	0h	Reserved

22.1.3.1.28 MCAN_RES03 Register (Offset = 23Ch) [reset = 0h]

MCAN_RES03 is shown in Figure 22-46 and described in Table 22-40.

Return to Summary Table.

RES03

Figure 22-46. MCAN_RES03 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES03																															
R-0h																															

Table 22-40. MCAN_RES03 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES03	R	0h	Reserved

22.1.3.1.29 MCAN_ECR Register (Offset = 240h) [reset = 0h]

MCAN_ECR is shown in Figure 22-47 and described in Table 22-41.

Return to Summary Table.

ECR

Figure 22-47. MCAN_ECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU25								CEL							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC							TEC							
R-0h	R-0h							R-0h							

Table 22-41. MCAN_ECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU25	R	0h	Reserved
23-16	CEL	R	0h	CAN Error Logging
15	RP	R	0h	Recieve Error Passive
14-8	REC	R	0h	Recieve Error Counter
7-0	TEC	R	0h	Transmit Error Counter

22.1.3.1.30 MCAN_PSR Register (Offset = 244h) [reset = 707h]

MCAN_PSR is shown in Figure 22-48 and described in Table 22-42.

Return to Summary Table.

PSR

Figure 22-48. MCAN_PSR Register

31	30	29	28	27	26	25	24
NU27							
R-0h							
23	22	21	20	19	18	17	16
NU27	TDCV						
R-0h	R-0h						
15	14	13	12	11	10	9	8
NU26	PXE	RFDF	RBRS	RESI	DLEC		
R-0h	R-0h	R-0h	R-0h	R-0h	R-7h		
7	6	5	4	3	2	1	0
BO	EW	EP	ACT		LEC		
R-0h	R-0h	R-0h	R-0h		R-7h		

Table 22-42. MCAN_PSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU27	R	0h	Reserved
22-16	TDCV	R	0h	Transmitter Delay Compensation Value
15	NU26	R	0h	Reserved
14	PXE	R	0h	Protocol Exception Event
13	RFDF	R	0h	Recieved a CAN FD Message
12	RBRS	R	0h	BRS flag of last recieved CAN FD Message
11	RESI	R	0h	ESI flag of last recieved CAN FD Message
10-8	DLEC	R	7h	Data Phase Last Error Code
7	BO	R	0h	Bus_Off status
6	EW	R	0h	Warning Status
5	EP	R	0h	Error Passive
4-3	ACT	R	0h	Activity
2-0	LEC	R	7h	Last Error Code

22.1.3.1.31 MCAN_TDCR Register (Offset = 248h) [reset = 0h]

MCAN_TDCR is shown in Figure 22-49 and described in Table 22-43.

Return to Summary Table.

TDCR

Figure 22-49. MCAN_TDCR Register

31	30	29	28	27	26	25	24
NU29							
R-0h							
23	22	21	20	19	18	17	16
NU29							
R-0h							
15	14	13	12	11	10	9	8
NU29	TDCO						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
NU28	TDCF						
R-0h	R/W-0h						

Table 22-43. MCAN_TDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	NU29	R	0h	Reserved
14-8	TDCO	R/W	0h	Transmitter Delay Compensation Offset
7	NU28	R	0h	Reserved
6-0	TDCF	R/W	0h	Transmitter Delay Compensation Filter Window Length

22.1.3.1.32 MCAN_RES04 Register (Offset = 24Ch) [reset = 0h]

MCAN_RES04 is shown in Figure 22-50 and described in Table 22-44.

Return to Summary Table.

RES04

Figure 22-50. MCAN_RES04 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES04																															
R-0h																															

Table 22-44. MCAN_RES04 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES04	R	0h	Reserved

22.1.3.1.33 MCAN_IR Register (Offset = 250h) [reset = 0h]

MCAN_IR is shown in Figure 22-51 and described in Table 22-45.

Return to Summary Table.

IR

Figure 22-51. MCAN_IR Register

31	30	29	28	27	26	25	24
NU30		ARA	PED	PEA	WDI	BO	EW
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-45. MCAN_IR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU30	R	0h	Reserved
29	ARA	R/W	0h	Access to Reserved Address
28	PED	R/W	0h	Protocol Error in data Phase
27	PEA	R/W	0h	Protocol Error in Arbitration Phase
26	WDI	R/W	0h	Watchdog Interrupt
25	BO	R/W	0h	Bus_Off Status
24	EW	R/W	0h	Warning Status
23	EP	R/W	0h	Error Passive
22	ELO	R/W	0h	Error Logging Overflow
21	BEU	R/W	0h	Bit Error Uncorrected
20	BEC	R/W	0h	Bit Error Corrected
19	DRX	R/W	0h	Message stored to Dedicated Rx Buffer
18	TOO	R/W	0h	Timeout Occurred
17	MRAF	R/W	0h	Message RAM Access Failure
16	TSW	R/W	0h	Timestamp Wraparound
15	TEFL	R/W	0h	Tx Event FIFO Element Lost
14	TEFF	R/W	0h	Tx Event FIFO Full
13	TEFW	R/W	0h	Tx Event FIFO Watermark Reached
12	TEFN	R/W	0h	Tx Event FIFO New Entry
11	TFE	R/W	0h	Tx FIFO Empty
10	TCF	R/W	0h	Transmission Cancellation Finished
9	TC	R/W	0h	Transmission Complete
8	HPM	R/W	0h	High Priority Message
7	RF1L	R/W	0h	Rx FIFO 1 Message Lost

Table 22-45. MCAN_IR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RF1F	R/W	0h	Rx FIFO 1 Full
5	RF1W	R/W	0h	Rx FIFO 1 Watermark Reached
4	RF1N	R/W	0h	Rx FIFO 1 New Message
3	RF0L	R/W	0h	Rx FIFO 0 Message Lost
2	RF0F	R/W	0h	Rx FIFO 0 Full
1	RF0W	R/W	0h	Rx FIFO 0 Watermark Reached
0	RF0N	R/W	0h	Rx FIFO 0 New Message

22.1.3.1.34 MCAN_IE Register (Offset = 254h) [reset = 0h]

MCAN_IE is shown in Figure 22-52 and described in Table 22-46.

Return to Summary Table.

IE

Figure 22-52. MCAN_IE Register

31	30	29	28	27	26	25	24
NU31		ARAE	PEDE	PEAE	WDIE	BOE	EWE
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPE	ELOE	BEUE	BECE	DRX	TOOE	MRAFE	TSWE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-46. MCAN_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU31	R	0h	Reserved
29	ARAE	R/W	0h	Accees to Reserve Address Interrupt Enable
28	PEDE	R/W	0h	Protocol Error in Data Phase Interrupt Enable
27	PEAE	R/W	0h	Protocol Error in Arbitration Phase Interrupt Enable
26	WDIE	R/W	0h	Watchdog Interrupt Enable
25	BOE	R/W	0h	Bus_Off Status Interrupt Enable
24	EWE	R/W	0h	Warning Status Interrupt Enable
23	EPE	R/W	0h	Error Passive Interrupt Enable
22	ELOE	R/W	0h	Error Logging Overflow Interrupt Enable
21	BEUE	R/W	0h	Bit Error Uncorrected Interrupt Enable
20	BECE	R/W	0h	Bit Error Corrected Interrupt Enable
19	DRX	R/W	0h	Message stored to Dedicated Rx Buffer Interrupt Enable
18	TOOE	R/W	0h	Timeout Occurred Interrupt Enable
17	MRAFE	R/W	0h	Message RAM Access Failure Interrupt Enable
16	TSWE	R/W	0h	Timestamp Wraparound Interrupt Enable
15	TEFLE	R/W	0h	Tx Event FIFO Event Lost Interrupt Enable
14	TEFFE	R/W	0h	Tx Event FIFO Full Interrupt Enable
13	TEFWE	R/W	0h	Tx Event FIFO Watermark Reached Interrupt enable
12	TEFNE	R/W	0h	Tx Event FIFO New Entry Interrupt Enable
11	TFEE	R/W	0h	Tx FIFO Empty Interrupt Enable
10	TCFE	R/W	0h	Transmission Cancellation Finishied Interrupt Enable
9	TCE	R/W	0h	Transmission Completed Interrupt Enable
8	HPME	R/W	0h	High Priority message Interrupt Enable
7	RF1LE	R/W	0h	rx FIFO 1 Message Lost Interrupt Enable

Table 22-46. MCAN_IE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RF1FE	R/W	0h	Rx FIFO 1 Full Interrupt Enable
5	RF1WE	R/W	0h	Rx FIFO 1 Watermark Reached Interrupt Enable
4	RF1NE	R/W	0h	Rx FIFO 1 New Message Interrupt Enable
3	RF0LE	R/W	0h	Rx FIFO 0 Message Lost Interrupt Enable
2	RF0FE	R/W	0h	Rx FIFO 0 Full Interrupt Enable
1	RF0WE	R/W	0h	Rx FIFO 0 Watermark Reached Interrupt Enable
0	RF0NE	R/W	0h	Rx FIFO 0 New Message Interrupt Enable

22.1.3.1.35 MCAN_ILS Register (Offset = 258h) [reset = 0h]

MCAN_ILS is shown in Figure 22-53 and described in Table 22-47.

Return to Summary Table.

ILS

Figure 22-53. MCAN_ILS Register

31	30	29	28	27	26	25	24
NU32		ARAL	PEDL	PEAL	WDIL	BOL	EWL
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-47. MCAN_ILS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU32	R	0h	Reserved
29	ARAL	R/W	0h	Accees to Reserve Address Interrupt Line
28	PEDL	R/W	0h	Protocol Error in Data Phase Interrupt Line
27	PEAL	R/W	0h	Protocol Error in Arbitration Phase Interrupt Line
26	WDIL	R/W	0h	Watchdog Interrupt Line
25	BOL	R/W	0h	Bus_Off Status Interrupt Line
24	EWL	R/W	0h	Warning Status Interrupt Line
23	EPL	R/W	0h	Error Passive Interrupt Line
22	ELOL	R/W	0h	Error Logging Overflow Interrupt Line
21	BEUL	R/W	0h	Bit Error Uncorrected Interrupt Line
20	BECL	R/W	0h	Bit Error Corrected Interrupt Line
19	DRXL	R/W	0h	Message stored to Dedicated Rx Buffer Interrupt Line
18	TOOL	R/W	0h	Timeout Occurred Interrupt Line
17	MRAFL	R/W	0h	Message RAM Access Failure Interrupt Line
16	TSWL	R/W	0h	Timestamp Wraparound Interrupt Line
15	TEFLL	R/W	0h	Tx Event FIFO Event Lost Interrupt Line
14	TEFFL	R/W	0h	Tx Event FIFO Full Interrupt Line
13	TEFWL	R/W	0h	Tx Event FIFO Watermark Reached Interrupt Line
12	TEFNL	R/W	0h	Tx Event FIFO New Entry Interrupt Line
11	TFEL	R/W	0h	Tx FIFO Empty Interrupt Line
10	TCFL	R/W	0h	Transmission Cancellation Finishied Interrupt Line
9	TCL	R/W	0h	Transmission Completed Interrupt Line
8	HPML	R/W	0h	High Priority message Interrupt Line
7	RF1LL	R/W	0h	rx FIFO 1 Message Lost Interrupt Line

Table 22-47. MCAN_ILS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RF1FL	R/W	0h	Rx FIFO 1 Full Interrupt Line
5	RF1WL	R/W	0h	Rx FIFO 1 Watermark Reached Interrupt Line
4	RF1NL	R/W	0h	Rx FIFO 1 New Message Interrupt Line
3	RF0LL	R/W	0h	Rx FIFO 0 Message Lost Interrupt Line
2	RF0FL	R/W	0h	Rx FIFO 0 Full Interrupt Line
1	RF0WL	R/W	0h	Rx FIFO 0 Watermark Reached Interrupt Line
0	RF0NL	R/W	0h	Rx FIFO 0 New Message Interrupt Line

22.1.3.1.36 MCAN_ILE Register (Offset = 25Ch) [reset = 0h]

MCAN_ILE is shown in Figure 22-54 and described in Table 22-48.

Return to Summary Table.

ILE

Figure 22-54. MCAN_ILE Register

31	30	29	28	27	26	25	24
NU33							
R-0h							
23	22	21	20	19	18	17	16
NU33							
R-0h							
15	14	13	12	11	10	9	8
NU33							
R-0h							
7	6	5	4	3	2	1	0
NU33						EINT1	EINT0
R-0h						R/W-0h	R/W-0h

Table 22-48. MCAN_ILE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU33	R	0h	Reserved
1	EINT1	R/W	0h	Enable Interrupt Line 1
0	EINT0	R/W	0h	Enable Interrupt Line 0

22.1.3.1.37 MCAN_RES05 Register (Offset = 260h) [reset = 0h]

MCAN_RES05 is shown in Figure 22-55 and described in Table 22-49.

Return to Summary Table.

RES05

Figure 22-55. MCAN_RES05 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES05																															
R-0h																															

Table 22-49. MCAN_RES05 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES05	R	0h	Reserved

22.1.3.1.38 MCAN_RES06 Register (Offset = 264h) [reset = 0h]

MCAN_RES06 is shown in Figure 22-56 and described in Table 22-50.

Return to Summary Table.

RES06

Figure 22-56. MCAN_RES06 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES06																															
R-0h																															

Table 22-50. MCAN_RES06 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES06	R	0h	Reserved

22.1.3.1.39 MCAN_RES07 Register (Offset = 268h) [reset = 0h]

MCAN_RES07 is shown in Figure 22-57 and described in Table 22-51.

Return to Summary Table.

RES07

Figure 22-57. MCAN_RES07 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES07																															
R-0h																															

Table 22-51. MCAN_RES07 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES07	R	0h	Reserved

22.1.3.1.40 MCAN_RES08 Register (Offset = 26Ch) [reset = 0h]

MCAN_RES08 is shown in Figure 22-58 and described in Table 22-52.

Return to Summary Table.

RES08

Figure 22-58. MCAN_RES08 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES08																															
R-0h																															

Table 22-52. MCAN_RES08 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES08	R	0h	Reserved

22.1.3.1.41 MCAN_RES09 Register (Offset = 270h) [reset = 0h]

MCAN_RES09 is shown in Figure 22-59 and described in Table 22-53.

Return to Summary Table.

RES09

Figure 22-59. MCAN_RES09 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES09																															
R-0h																															

Table 22-53. MCAN_RES09 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES09	R	0h	Reserved

22.1.3.1.42 MCAN_RES10 Register (Offset = 274h) [reset = 0h]

MCAN_RES10 is shown in Figure 22-60 and described in Table 22-54.

Return to Summary Table.

RES10

Figure 22-60. MCAN_RES10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES10																															
R-0h																															

Table 22-54. MCAN_RES10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES10	R	0h	Reserved

22.1.3.1.43 MCAN_RES11 Register (Offset = 278h) [reset = 0h]

MCAN_RES11 is shown in Figure 22-61 and described in Table 22-55.

Return to Summary Table.

RES11

Figure 22-61. MCAN_RES11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES11																															
R-0h																															

Table 22-55. MCAN_RES11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES11	R	0h	Reserved

22.1.3.1.44 MCAN_RES12 Register (Offset = 27Ch) [reset = 0h]

MCAN_RES12 is shown in Figure 22-62 and described in Table 22-56.

Return to Summary Table.

RES12

Figure 22-62. MCAN_RES12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES12																															
R-0h																															

Table 22-56. MCAN_RES12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES12	R	0h	Reserved

22.1.3.1.45 MCAN_GFC Register (Offset = 280h) [reset = 0h]

MCAN_GFC is shown in Figure 22-63 and described in Table 22-57.

Return to Summary Table.

GFC

Figure 22-63. MCAN_GFC Register

31	30	29	28	27	26	25	24
NU34							
R-0h							
23	22	21	20	19	18	17	16
NU34							
R-0h							
15	14	13	12	11	10	9	8
NU34							
R-0h							
7	6	5	4	3	2	1	0
NU34		ANFS		ANFE		RRFS	RRFE
R-0h		R/W-0h		R/W-0h		R/W-0h	R/W-0h

Table 22-57. MCAN_GFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU34	R	0h	Reserved
5-4	ANFS	R/W	0h	Accept Non-matching Frames Standard
3-2	ANFE	R/W	0h	Accept Non-matching Frames Extended
1	RRFS	R/W	0h	reject Remote Frames Standard
0	RRFE	R/W	0h	reject Remote Frames Extended

22.1.3.1.46 MCAN_SIDFC Register (Offset = 284h) [reset = 0h]

MCAN_SIDFC is shown in Figure 22-64 and described in Table 22-58.

Return to Summary Table.

SIDFC

Figure 22-64. MCAN_SIDFC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU36								LSS_S							
R-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLSSA_S														NU35	
R/W-0h														R-0h	

Table 22-58. MCAN_SIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU36	R	0h	Reserved
23-16	LSS_S	R/W	0h	List Size Standard
15-2	FLSSA_S	R/W	0h	Filter List Standard Start Address
1-0	NU35	R	0h	Reserved

22.1.3.1.47 MCAN_XIDFC Register (Offset = 288h) [reset = 0h]

MCAN_XIDFC is shown in Figure 22-65 and described in Table 22-59.

Return to Summary Table.

XIDFC

Figure 22-65. MCAN_XIDFC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU38								LSS_X							
R-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLSSA_X														NU37	
R/W-0h														R-0h	

Table 22-59. MCAN_XIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU38	R	0h	Reserved
23-16	LSS_X	R/W	0h	List Size Standard
15-2	FLSSA_X	R/W	0h	Filter List Standard Start Address
1-0	NU37	R	0h	Reserved

22.1.3.1.48 MCAN_RES13 Register (Offset = 28Ch) [reset = 0h]

MCAN_RES13 is shown in Figure 22-66 and described in Table 22-60.

Return to Summary Table.

RES13

Figure 22-66. MCAN_RES13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES13																															
R-0h																															

Table 22-60. MCAN_RES13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES13	R	0h	Reserved

22.1.3.1.49 MCAN_XIDAM Register (Offset = 290h) [reset = 1FFFFFFFh]

MCAN_XIDAM is shown in Figure 22-67 and described in Table 22-61.

Return to Summary Table.

XIDAM

Figure 22-67. MCAN_XIDAM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU39			EIDM												
R-0h			R/W-1FFFFFFFh												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIDM															
R/W-1FFFFFFFh															

Table 22-61. MCAN_XIDAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU39	R	0h	Reserved
28-0	EIDM	R/W	1FFFFFFFh	Extended ID Mask

22.1.3.1.50 MCAN_HPMS Register (Offset = 294h) [reset = 0h]

MCAN_HPMS is shown in Figure 22-68 and described in Table 22-62.

Return to Summary Table.

HPMS

Figure 22-68. MCAN_HPMS Register

31	30	29	28	27	26	25	24
NU40							
R-0h							
23	22	21	20	19	18	17	16
NU40							
R-0h							
15	14	13	12	11	10	9	8
FLST	FIDX						
R-0h	R-0h						
7	6	5	4	3	2	1	0
MSI		BIDX					
R-0h		R-0h					

Table 22-62. MCAN_HPMS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU40	R	0h	Reserved
15	FLST	R	0h	Filter List
14-8	FIDX	R	0h	Filter Index
7-6	MSI	R	0h	Message Storeage Indicator
5-0	BIDX	R	0h	Buffer Index

22.1.3.1.51 MCAN_NDAT1 Register (Offset = 298h) [reset = 0h]

MCAN_NDAT1 is shown in Figure 22-69 and described in Table 22-63.

Return to Summary Table.

NDAT1

Figure 22-69. MCAN_NDAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND0_31																															
R/W-0h																															

Table 22-63. MCAN_NDAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ND0_31	R/W	0h	New Data 0-31

22.1.3.1.52 MCAN_NDAT2 Register (Offset = 29Ch) [reset = 0h]

MCAN_NDAT2 is shown in Figure 22-70 and described in Table 22-64.

Return to Summary Table.

NDAT2

Figure 22-70. MCAN_NDAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND32_63																															
R/W-0h																															

Table 22-64. MCAN_NDAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ND32_63	R/W	0h	New Data 32-63

22.1.3.1.53 MCAN_RXF0C Register (Offset = 2A0h) [reset = 0h]

MCAN_RXF0C is shown in Figure 22-71 and described in Table 22-65.

Return to Summary Table.

RXF0C

Figure 22-71. MCAN_RXF0C Register

31	30	29	28	27	26	25	24
F0OM	F0WM						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
NU42	F0S						
R-0h	R/W-0h						
15	14	13	12	11	10	9	8
F0S	F0SA						
R/W-0h	R/W-0h						
7	6	5	4	3	2	1	0
F0SA						NU41	
R/W-0h						R-0h	

Table 22-65. MCAN_RXF0C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F0OM	R/W	0h	Rx FIFO 0 Operation Mode
30-24	F0WM	R/W	0h	Rx FIFO 0 Watermark
23	NU42	R	0h	Reserved
22-15	F0S	R/W	0h	Rx FIFO 0 Size
14-2	F0SA	R/W	0h	Rx FIFO 0 Start Address
1-0	NU41	R	0h	Reserved

22.1.3.1.54 MCAN_RXF0S Register (Offset = 2A4h) [reset = 0h]

MCAN_RXF0S is shown in Figure 22-72 and described in Table 22-66.

Return to Summary Table.

RXF0S

Figure 22-72. MCAN_RXF0S Register

31	30	29	28	27	26	25	24
NU46						RF0L	F0F
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
NU45		F0PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
NU44		F0GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
NU43		F0FL					
R-0h		R-0h					

Table 22-66. MCAN_RXF0S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU46	R	0h	Reserved
25	RF0L	R	0h	Rx FIFO 0 Message Lost
24	F0F	R	0h	Rx FIFO 0 Full
23-22	NU45	R	0h	Reserved
21-16	F0PI	R	0h	Rx FIFO 0 Put Index
15-14	NU44	R	0h	Reserved
13-8	F0GI	R	0h	Rx FIFO 0 Get Index
7-6	NU43	R	0h	Reserved
5-0	F0FL	R	0h	Rx FIFO 0 Fill Level

22.1.3.1.55 MCAN_RXF0A Register (Offset = 2A8h) [reset = 0h]

MCAN_RXF0A is shown in Figure 22-73 and described in Table 22-67.

Return to Summary Table.

RXF0A

Figure 22-73. MCAN_RXF0A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU47																										F0AI					
R-0h																										R/W-0h					

Table 22-67. MCAN_RXF0A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU47	R	0h	Reserved
5-0	F0AI	R/W	0h	Rx FIFO 0 Acknowledge Index

22.1.3.1.56 MCAN_RXBC Register (Offset = 2ACh) [reset = 0h]

MCAN_RXBC is shown in Figure 22-74 and described in Table 22-68.

Return to Summary Table.

RXBC

Figure 22-74. MCAN_RXBC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU49															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBSA														NU48	
R/W-0h														R-0h	

Table 22-68. MCAN_RXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU49	R	0h	Reserved
15-2	RBSA	R/W	0h	Rx Buffer Start Address
1-0	NU48	R	0h	Reserved

22.1.3.1.57 MCAN_RXF1C Register (Offset = 2B0h) [reset = 0h]

MCAN_RXF1C is shown in Figure 22-75 and described in Table 22-69.

Return to Summary Table.

RXF1C

Figure 22-75. MCAN_RXF1C Register

31	30	29	28	27	26	25	24
F1OM	F1WM						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
NU50	F1S						
R-0h	R/W-0h						
15	14	13	12	11	10	9	8
F1S	F1SA						
R/W-0h	R/W-0h						
7	6	5	4	3	2	1	0
F1SA						NU499	
R/W-0h						R-0h	

Table 22-69. MCAN_RXF1C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F1OM	R/W	0h	Rx FIFO 0 Operation Mode
30-24	F1WM	R/W	0h	Rx FIFO 0 Watermark
23	NU50	R	0h	Reserved
22-15	F1S	R/W	0h	Rx FIFO 0 Size
14-2	F1SA	R/W	0h	Rx FIFO 0 Start Address
1-0	NU499	R	0h	Reserved

22.1.3.1.58 MCAN_RXF1S Register (Offset = 2B4h) [reset = 0h]

MCAN_RXF1S is shown in Figure 22-76 and described in Table 22-70.

Return to Summary Table.

RXF1S

Figure 22-76. MCAN_RXF1S Register

31	30	29	28	27	26	25	24
NU54						RF1L	F1F
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
NU53		F1PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
NU52		F1GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
NU51		F1FL					
R-0h		R-0h					

Table 22-70. MCAN_RXF1S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU54	R	0h	Reserved
25	RF1L	R	0h	Rx FIFO 0 Message Lost
24	F1F	R	0h	Rx FIFO 0 Full
23-22	NU53	R	0h	Reserved
21-16	F1PI	R	0h	Rx FIFO 0 Put Index
15-14	NU52	R	0h	Reserved
13-8	F1GI	R	0h	Rx FIFO 0 Get Index
7-6	NU51	R	0h	Reserved
5-0	F1FL	R	0h	Rx FIFO 0 Fill Level

22.1.3.1.59 MCAN_RXF1A Register (Offset = 2B8h) [reset = 0h]

MCAN_RXF1A is shown in Figure 22-77 and described in Table 22-71.

Return to Summary Table.

RXF1A

Figure 22-77. MCAN_RXF1A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU55																										F1AI					
R-0h																										R/W-0h					

Table 22-71. MCAN_RXF1A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU55	R	0h	Reserved
5-0	F1AI	R/W	0h	Rx FIFO 0 Acknowledge Index

22.1.3.1.60 MCAN_RXESC Register (Offset = 2BCh) [reset = 0h]

MCAN_RXESC is shown in Figure 22-78 and described in Table 22-72.

Return to Summary Table.

RXESC

Figure 22-78. MCAN_RXESC Register

31	30	29	28	27	26	25	24
NU58							
R-0h							
23	22	21	20	19	18	17	16
NU58							
R-0h							
15	14	13	12	11	10	9	8
NU58					RBDS		
R-0h					R/W-0h		
7	6	5	4	3	2	1	0
NU57	F1DS			NU56	F0DS		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 22-72. MCAN_RXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU58	R	0h	Reserved
10-8	RBDS	R/W	0h	Rx Buffer data Field Size
7	NU57	R	0h	Reserved
6-4	F1DS	R/W	0h	Rx FIFO 1 Data Field Size
3	NU56	R	0h	Reserved
2-0	F0DS	R/W	0h	Rx FIFO 0 Data Field Size

22.1.3.1.61 MCAN_TXBC Register (Offset = 2C0h) [reset = 0h]

MCAN_TXBC is shown in Figure 22-79 and described in Table 22-73.

Return to Summary Table.

TXBC

Figure 22-79. MCAN_TXBC Register

31	30	29	28	27	26	25	24
NU61	TFQM	TFQS					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
NU60		NDTB					
R-0h		R-0h					
15	14	13	12	11	10	9	8
TBSA							
R-0h							
7	6	5	4	3	2	1	0
TBSA						NU59	
R-0h						R-0h	

Table 22-73. MCAN_TXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU61	R	0h	Reserved
30	TFQM	R	0h	Tx FIFO/Queue Mode
29-24	TFQS	R	0h	Transmit FIFO/Queue Size
23-22	NU60	R	0h	Reserved
21-16	NDTB	R	0h	Number of Dedicated Transmit Buffers
15-2	TBSA	R	0h	Tx Buffers Start Address
1-0	NU59	R	0h	Reserved

22.1.3.1.62 MCAN_TXFQS Register (Offset = 2C4h) [reset = 0h]

MCAN_TXFQS is shown in Figure 22-80 and described in Table 22-74.

Return to Summary Table.

TXFQS

Figure 22-80. MCAN_TXFQS Register

31	30	29	28	27	26	25	24
NU64							
R-0h							
23	22	21	20	19	18	17	16
NU64		TFQF	TFQPI				
R-0h		R-0h	R-0h				
15	14	13	12	11	10	9	8
NU63			TFGI				
R-0h			R-0h				
7	6	5	4	3	2	1	0
NU62		TFFL					
R-0h		R-0h					

Table 22-74. MCAN_TXFQS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	NU64	R	0h	Reserved
21	TFQF	R	0h	Tx FIFO/Queue Full
20-16	TFQPI	R	0h	Tx FIFO/Queue Put Index
15-13	NU63	R	0h	Reserved
12-8	TFGI	R	0h	Tx Queue Get Index
7-6	NU62	R	0h	Reserved
5-0	TFFL	R	0h	Tx FIFO Free Level

22.1.3.1.63 MCAN_TXESC Register (Offset = 2C8h) [reset = 0h]

MCAN_TXESC is shown in Figure 22-81 and described in Table 22-75.

Return to Summary Table.

TXESC

Figure 22-81. MCAN_TXESC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU65															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU65													TBDS		
R-0h													R/W-0h		

Table 22-75. MCAN_TXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU65	R	0h	Reserved
2-0	TBDS	R/W	0h	Tx Buffer Data Field Size

22.1.3.1.64 MCAN_TXBRP Register (Offset = 2CCh) [reset = 0h]

MCAN_TXBRP is shown in Figure 22-82 and described in Table 22-76.

Return to Summary Table.

TXBRP

Figure 22-82. MCAN_TXBRP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP																															
R-0h																															

Table 22-76. MCAN_TXBRP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TRP	R	0h	Transmission Request Pending

22.1.3.1.65 MCAN_TXBAR Register (Offset = 2D0h) [reset = 0h]

MCAN_TXBAR is shown in Figure 22-83 and described in Table 22-77.

Return to Summary Table.

TXBAR

Figure 22-83. MCAN_TXBAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR																															
R/W0C-0h																															

Table 22-77. MCAN_TXBAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	AR	R/W0C	0h	Add request

22.1.3.1.66 MCAN_TXBCR Register (Offset = 2D4h) [reset = 0h]

MCAN_TXBCR is shown in Figure 22-84 and described in Table 22-78.

Return to Summary Table.

TXBCR

Figure 22-84. MCAN_TXBCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR																															
R/W0C-0h																															

Table 22-78. MCAN_TXBCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CR	R/W0C	0h	Cancellation Request

22.1.3.1.67 MCAN_TXBTO Register (Offset = 2D8h) [reset = 0h]

MCAN_TXBTO is shown in Figure 22-85 and described in Table 22-79.

Return to Summary Table.

TXBTO

Figure 22-85. MCAN_TXBTO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO																															
R-0h																															

Table 22-79. MCAN_TXBTO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TO	R	0h	Transmission Occurred

22.1.3.1.68 MCAN_TXBCF Register (Offset = 2DCh) [reset = 0h]

MCAN_TXBCF is shown in Figure 22-86 and described in Table 22-80.

Return to Summary Table.

TXBCF

Figure 22-86. MCAN_TXBCF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF																															
R-0h																															

Table 22-80. MCAN_TXBCF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CF	R	0h	Cancellation Finished

22.1.3.1.69 MCAN_TXBTIE Register (Offset = 2E0h) [reset = 0h]

MCAN_TXBTIE is shown in Figure 22-87 and described in Table 22-81.

Return to Summary Table.

TXBTIE

Figure 22-87. MCAN_TXBTIE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE																															
R/W-0h																															

Table 22-81. MCAN_TXBTIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIE	R/W	0h	Transmission Interrupt Enable

22.1.3.1.70 MCAN_TXBCIE Register (Offset = 2E4h) [reset = 0h]

MCAN_TXBCIE is shown in Figure 22-88 and described in Table 22-82.

Return to Summary Table.

TXBCIE

Figure 22-88. MCAN_TXBCIE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE																															
R/W-0h																															

Table 22-82. MCAN_TXBCIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFIE	R/W	0h	Cancellation Finished Interrupt Enable

22.1.3.1.71 MCAN_RES14 Register (Offset = 2E8h) [reset = 0h]

MCAN_RES14 is shown in Figure 22-89 and described in Table 22-83.

Return to Summary Table.

RES14

Figure 22-89. MCAN_RES14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES14																															
R-0h																															

Table 22-83. MCAN_RES14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES14	R	0h	Reserved

22.1.3.1.72 MCAN_RES15 Register (Offset = 2ECh) [reset = 0h]

MCAN_RES15 is shown in Figure 22-90 and described in Table 22-84.

Return to Summary Table.

RES15

Figure 22-90. MCAN_RES15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES15																															
R-0h																															

Table 22-84. MCAN_RES15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES15	R	0h	Reserved

22.1.3.1.73 MCAN_TXEFC Register (Offset = 2F0h) [reset = 0h]

MCAN_TXEFC is shown in Figure 22-91 and described in Table 22-85.

Return to Summary Table.

TXEFC

Figure 22-91. MCAN_TXEFC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU68		EFWM						NU67		EFS					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFSA														NU66	
R/W-0h														R/W-0h	

Table 22-85. MCAN_TXEFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU68	R/W	0h	Reserved
29-24	EFWM	R/W	0h	Event FIFO Watermark
23-22	NU67	R/W	0h	Reserved
21-16	EFS	R/W	0h	Event FIFO Size
15-2	EFSA	R/W	0h	Event FIFO Start Address
1-0	NU66	R/W	0h	Reserved

22.1.3.1.74 MCAN_TXEFS Register (Offset = 2F4h) [reset = 0h]

MCAN_TXEFS is shown in Figure 22-92 and described in Table 22-86.

Return to Summary Table.

TXEFS

Figure 22-92. MCAN_TXEFS Register

31	30	29	28	27	26	25	24
NU72						TEFL	EFF
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
NU71			EFPI				
R-0h			R-0h				
15	14	13	12	11	10	9	8
NU70			EFGI				
R-0h			R-0h				
7	6	5	4	3	2	1	0
NU69		EFFL					
R-0h		R-0h					

Table 22-86. MCAN_TXEFS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU72	R	0h	Reserved
25	TEFL	R	0h	Tx Event FIFO Element Lost
24	EFF	R	0h	Event FIFO Full
23-21	NU71	R	0h	Reserved
20-16	EFPI	R	0h	Event FIFO Put Index
15-13	NU70	R	0h	Reserved
12-8	EFGI	R	0h	Event FIFO Get Index
7-6	NU69	R	0h	Reserved
5-0	EFFL	R	0h	Event FIFO FIll Level

22.1.3.1.75 MCAN_TXEFA Register (Offset = 2F8h) [reset = 0h]

MCAN_TXEFA is shown in Figure 22-93 and described in Table 22-87.

Return to Summary Table.

TXEFA

Figure 22-93. MCAN_TXEFA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU73																											EFAI			
R-0h																											R-0h			

Table 22-87. MCAN_TXEFA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	NU73	R	0h	Reserved
4-0	EFAI	R	0h	Event FIFO Acknowledge Index

22.1.3.1.76 MCAN_RES16 Register (Offset = 2FCh) [reset = 0h]

MCAN_RES16 is shown in Figure 22-94 and described in Table 22-88.

Return to Summary Table.

RES16

Figure 22-94. MCAN_RES16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES16																															
R-0h																															

Table 22-88. MCAN_RES16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES16	R	0h	Reserved

22.1.3.2 MSS_MCAN_ECC Registers

Table 22-89 lists the memory-mapped registers for the MSS_MCAN_ECC. All register offset addresses not listed in Table 22-89 should be considered as reserved locations and the register contents should not be modified.

Table 22-89. MSS_MCAN_ECC Registers

Offset	Acronym	Register Name	Section
0h	MCANSS_ECC_AGGR_REVISION	MCANSS_ECC_AGGR_REVISION	Section 22.1.3.2.1
8h	MCANSS_ECC_VECTOR	MCANSS_ECC_VECTOR	Section 22.1.3.2.2
Ch	MCANSS_ECC_MISC_STATUS	MCANSS_ECC_MISC_STATUS	Section 22.1.3.2.3
10h	MCANSS_ECC_WRAP_REVISION	MCANSS_ECC_WRAP_REVISION	Section 22.1.3.2.4
14h	MCANSS_ECC_CONTROL	MCANSS_ECC_CONTROL	Section 22.1.3.2.5
18h	MCANSS_ECC_ERR_CTRL1	MCANSS_ECC_ERR_CTRL1	Section 22.1.3.2.6
1Ch	MCANSS_ECC_ERR_CTRL2	MCANSS_ECC_ERR_CTRL2	Section 22.1.3.2.7
20h	MCANSS_ECC_ERR_STAT1	MCANSS_ECC_ERR_STAT1	Section 22.1.3.2.8
24h	MCANSS_ECC_ERR_STAT2	MCANSS_ECC_ERR_STAT2	Section 22.1.3.2.9
3Ch	MCANSS_ECC_SEC_EOI_REG	MCANSS_ECC_SEC_EOI_REG	Section 22.1.3.2.10
40h	MCANSS_ECC_SEC_STATUS_REG0	MCANSS_ECC_SEC_STATUS_REG0	Section 22.1.3.2.11
80h	MCANSS_ECC_SEC_ENABLE_SET_REG0	MCANSS_ECC_SEC_ENABLE_SET_REG0	Section 22.1.3.2.12
C0h	MCANSS_ECC_SEC_ENABLE_CLR_REG0	MCANSS_ECC_SEC_ENABLE_CLR_REG0	Section 22.1.3.2.13
13Ch	MCANSS_ECC_DED_EOI_REG	MCANSS_ECC_DED_EOI_REG	Section 22.1.3.2.14
140h	MCANSS_ECC_DED_STATUS_REG0	MCANSS_ECC_DED_STATUS_REG0	Section 22.1.3.2.15
180h	MCANSS_ECC_DED_ENABLE_SET_REG0	MCANSS_ECC_DED_ENABLE_SET_REG0	Section 22.1.3.2.16
1C0h	MCANSS_ECC_DED_ENABLE_CLR_REG0	MCANSS_ECC_DED_ENABLE_CLR_REG0	Section 22.1.3.2.17

Complex bit access types are encoded to fit into small table cells. Table 22-90 shows the codes that are used for access types in this section.

Table 22-90. MSS_MCAN_ECC Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W W	Write Write
W0C	W 0C	Write 0 to clear

22.1.3.2.1 MCANSS_ECC_AGGR_REVISION Register (Offset = 0h) [reset = 66A00B00h]

MCANSS_ECC_AGGR_REVISION is shown in Figure 22-95 and described in Table 22-91.

Return to Summary Table.

REV

Figure 22-95. MCANSS_ECC_AGGR_REVISION Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		BU		MODULE_ID											
R-1h		R-2h		R-6A0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVRTL					REVMAJ			CUSTOM		REVMIN					
R-1h					R-3h			R-0h		R-0h					

Table 22-91. MCANSS_ECC_AGGR_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Scheme
29-28	BU	R	2h	bu
27-16	MODULE_ID	R	6A0h	Module ID
15-11	REVRTL	R	1h	RTL version
10-8	REVMAJ	R	3h	Major version
7-6	CUSTOM	R	0h	Custom version
5-0	REVMIN	R	0h	Minor version

22.1.3.2.2 MCANSS_ECC_VECTOR Register (Offset = 8h) [reset = 0h]

MCANSS_ECC_VECTOR is shown in Figure 22-96 and described in Table 22-92.

Return to Summary Table.

VECTOR

Figure 22-96. MCANSS_ECC_VECTOR Register

31	30	29	28	27	26	25	24
NU1							RD_SVBUS_DONE
R-0h							R-0h
23	22	21	20	19	18	17	16
RD_SVBUS_ADDR							
R/W-0h							
15	14	13	12	11	10	9	8
RD_SVBUS	NU0				ECC_VEC		
R/W-0h	R-0h				R/W-0h		
7	6	5	4	3	2	1	0
ECC_VEC							
R/W-0h							

Table 22-92. MCANSS_ECC_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU1	R	0h	Reserved
24	RD_SVBUS_DONE	R	0h	Status to indicate if read on serial VBUS is complete
23-16	RD_SVBUS_ADDR	R/W	0h	Read address
15	RD_SVBUS	R/W	0h	Write 1 to trigger a read on the serial VBUS
14-11	NU0	R	0h	Reserved
10-0	ECC_VEC	R/W	0h	Value written to select the corresponding ECC RAM for control or status

22.1.3.2.3 MCANSS_ECC_MISC_STATUS Register (Offset = Ch) [reset = 1h]

MCANSS_ECC_MISC_STATUS is shown in Figure 22-97 and described in Table 22-93.

Return to Summary Table.

STAT

Figure 22-97. MCANSS_ECC_MISC_STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2																					NUM_RAMS										
R-0h																					R-1h										

Table 22-93. MCANSS_ECC_MISC_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU2	R	0h	Reserved
10-0	NUM_RAMS	R	1h	Indicates the number of RAMS serviced by the ECC aggregator

22.1.3.2.4 MCANSS_ECC_WRAP_REVISION Register (Offset = 10h) [reset = 66A40900h]

MCANSS_ECC_WRAP_REVISION is shown in Figure 22-98 and described in Table 22-94.

Return to Summary Table.

WRAP_REV

Figure 22-98. MCANSS_ECC_WRAP_REVISION Register

31	30	29	28	27	26	25	24
W_SCHEME		W_BU		W_MODULE_ID			
R-1h		R-2h		R-6A4h			
23	22	21	20	19	18	17	16
W_MODULE_ID							
R-6A4h							
15	14	13	12	11	10	9	8
W_REVRTL					W_REVMAJ		
R-1h					R-1h		
7	6	5	4	3	2	1	0
W_CUSTOM		RESERVED					
R-0h		R-					

Table 22-94. MCANSS_ECC_WRAP_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	W_SCHEME	R	1h	Scheme
29-28	W_BU	R	2h	bu
27-16	W_MODULE_ID	R	6A4h	Module ID
15-11	W_REVRTL	R	1h	RTL version
10-8	W_REVMAJ	R	1h	Major version
7-6	W_CUSTOM	R	0h	Custom version
5-0	RESERVED	R		

22.1.3.2.5 MCANSS_ECC_CONTROL Register (Offset = 14h) [reset = 7h]

MCANSS_ECC_CONTROL is shown in Figure 22-99 and described in Table 22-95.

Return to Summary Table.

CTRL

Figure 22-99. MCANSS_ECC_CONTROL Register

31	30	29	28	27	26	25	24
NU3							
R-0h							
23	22	21	20	19	18	17	16
NU3							
R-0h							
15	14	13	12	11	10	9	8
NU3							
R-0h							
7	6	5	4	3	2	1	0
NU3	ERROR_ONCE	FORCE_N_ROW	FORCE_DED	FORCE_SEC	EN_RMW	ECC_CHK	ECC_EN
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h

Table 22-95. MCANSS_ECC_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	NU3	R	0h	Reserved
6	ERROR_ONCE	R/W	0h	Force Error only once
5	FORCE_N_ROW	R/W	0h	Force Error on any RAM read
4	FORCE_DED	R/W	0h	Force Double Bit Error
3	FORCE_SEC	R/W	0h	Force Single Bit Error
2	EN_RMW	R/W	1h	Enable rmw
1	ECC_CHK	R/W	1h	Enable ECC check
0	ECC_EN	R/W	1h	Enable ECC

22.1.3.2.6 MCANSS_ECC_ERR_CTRL1 Register (Offset = 18h) [reset = 0h]

MCANSS_ECC_ERR_CTRL1 is shown in Figure 22-100 and described in Table 22-96.

Return to Summary Table.

ERR_CTRL1

Figure 22-100. MCANSS_ECC_ERR_CTRL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BIT1																ECC_ROW															
R/W-0h																R/W-0h															

Table 22-96. MCANSS_ECC_ERR_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT1	R/W	0h	Data bit that needs to be flipped when force_sec is set
15-0	ECC_ROW	R/W	0h	Row address where single or double-bit error needs to be applied. This is ignored if force_n_row is set

22.1.3.2.7 MCANSS_ECC_ERR_CTRL2 Register (Offset = 1Ch) [reset = 0h]

MCANSS_ECC_ERR_CTRL2 is shown in Figure 22-101 and described in Table 22-97.

Return to Summary Table.

ERR_CTRL2

Figure 22-101. MCANSS_ECC_ERR_CTRL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU4																ECC_BIT2															
R-0h																R/W-0h															

Table 22-97. MCANSS_ECC_ERR_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU4	R	0h	Reserved
15-0	ECC_BIT2	R/W	0h	Data bit that needs to be flipped if double bit error needs to be forced

22.1.3.2.8 MCANSS_ECC_ERR_STAT1 Register (Offset = 20h) [reset = 0h]

MCANSS_ECC_ERR_STAT1 is shown in Figure 22-102 and described in Table 22-98.

Return to Summary Table.

ERR_STAT1

Figure 22-102. MCANSS_ECC_ERR_STAT1 Register

31	30	29	28	27	26	25	24
ecc_row							
R-0h							
23	22	21	20	19	18	17	16
ecc_row							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						clr_ecc_ded	clr_ecc_sec
R-						R/W0C-0h	R/W0C-0h
7	6	5	4	3	2	1	0
RESERVED						ecc_ded	ecc_sec
R-						R/W0C-0h	R/W0C-0h

Table 22-98. MCANSS_ECC_ERR_STAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ecc_row	R	0h	Row address where the single or double-bit error has occurred
15-10	RESERVED	R		
9	clr_ecc_ded	R/W0C	0h	Clear Double Bit Error Status
8	clr_ecc_sec	R/W0C	0h	Clear Single Bit Error Status
7-2	RESERVED	R		
1	ecc_ded	R/W0C	0h	Level Double Bit Error Status
0	ecc_sec	R/W0C	0h	Level Single Bit Error Status

22.1.3.2.9 MCANSS_ECC_ERR_STAT2 Register (Offset = 24h) [reset = 0h]

MCANSS_ECC_ERR_STAT2 is shown in Figure 22-103 and described in Table 22-99.

Return to Summary Table.

ERR_STAT2

Figure 22-103. MCANSS_ECC_ERR_STAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BIT2_STS																ECC_BIT1_STS															
R-0h																R-0h															

Table 22-99. MCANSS_ECC_ERR_STAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT2_STS	R	0h	Data bit that corresponds to the double-bit error
15-0	ECC_BIT1_STS	R	0h	Data bit that corresponds to the single-bit error

22.1.3.2.10 MCANSS_ECC_SEC_EOI_REG Register (Offset = 3Ch) [reset = 0h]

MCANSS_ECC_SEC_EOI_REG is shown in Figure 22-104 and described in Table 22-100.

Return to Summary Table.

SEC_EOI_REG

Figure 22-104. MCANSS_ECC_SEC_EOI_REG Register

Table 22-100. MCANSS_ECC_SEC_EOI_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SEC_EOI_WR	R/W	0h	EOI Register

22.1.3.2.11 MCANSS_ECC_SEC_STATUS_REG0 Register (Offset = 40h) [reset = 0h]

MCANSS_ECC_SEC_STATUS_REG0 is shown in Figure 22-105 and described in Table 22-101.

Return to Summary Table.

SEC_STATUS_REG0

Figure 22-105. MCANSS_ECC_SEC_STATUS_REG0 Register

Table 22-101. MCANSS_ECC_SEC_STATUS_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SEC_PEND	R/W0C	0h	Interrupt Pending Status for msgmem_pend

22.1.3.2.12 MCANSS_ECC_SEC_ENABLE_SET_REG0 Register (Offset = 80h) [reset = 0h]

MCANSS_ECC_SEC_ENABLE_SET_REG0 is shown in Figure 22-106 and described in Table 22-102.

Return to Summary Table.

SEC_ENABLE_SET_REG0

Figure 22-106. MCANSS_ECC_SEC_ENABLE_SET_REG0 Register

Table 22-102. MCANSS_ECC_SEC_ENABLE_SET_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SEC_EN_SET	R/W0C	0h	Interrupt Enable Set Register for msgmem_pend

22.1.3.2.13 MCANSS_ECC_SEC_ENABLE_CLR_REG0 Register (Offset = C0h) [reset = 0h]

MCANSS_ECC_SEC_ENABLE_CLR_REG0 is shown in Figure 22-107 and described in Table 22-103.

Return to Summary Table.

SEC_ENABLE_CLR_REG0

Figure 22-107. MCANSS_ECC_SEC_ENABLE_CLR_REG0 Register

Table 22-103. MCANSS_ECC_SEC_ENABLE_CLR_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SEC_EN_CLR	R/W0C	0h	Interrupt Enable Clear Register for msgmem_pend

22.1.3.2.14 MCANSS_ECC_DED_EOI_REG Register (Offset = 13Ch) [reset = 0h]

MCANSS_ECC_DED_EOI_REG is shown in Figure 22-108 and described in Table 22-104.

Return to Summary Table.

DED_EOI_REG

Figure 22-108. MCANSS_ECC_DED_EOI_REG Register

Table 22-104. MCANSS_ECC_DED_EOI_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
0	DED_EOI_WR	R/W	0h	EOI Register

22.1.3.2.15 MCANSS_ECC_DED_STATUS_REG0 Register (Offset = 140h) [reset = 0h]

MCANSS_ECC_DED_STATUS_REG0 is shown in Figure 22-109 and described in Table 22-105.

Return to Summary Table.

DED_STATUS_REG0

Figure 22-109. MCANSS_ECC_DED_STATUS_REG0 Register

Table 22-105. MCANSS_ECC_DED_STATUS_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
0	DED_PEND	R/W0C	0h	Interrupt Pending Status for msgmem_pend

22.1.3.2.16 MCANSS_ECC_DED_ENABLE_SET_REG0 Register (Offset = 180h) [reset = 0h]

MCANSS_ECC_DED_ENABLE_SET_REG0 is shown in Figure 22-110 and described in Table 22-106.

Return to Summary Table.

DED_ENABLE_SET_REG0

Figure 22-110. MCANSS_ECC_DED_ENABLE_SET_REG0 Register

Table 22-106. MCANSS_ECC_DED_ENABLE_SET_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
0	DED_EN_SET	R/W0C	0h	Interrupt Enable Set Register for msgmem_pend

22.1.3.2.17 MCANSS_ECC_DED_ENABLE_CLR_REG0 Register (Offset = 1C0h) [reset = 0h]

MCANSS_ECC_DED_ENABLE_CLR_REG0 is shown in Figure 22-111 and described in Table 22-107.

Return to Summary Table.

DED_ENABLE_CLR_REG0

Figure 22-111. MCANSS_ECC_DED_ENABLE_CLR_REG0 Register

Table 22-107. MCANSS_ECC_DED_ENABLE_CLR_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
0	DED_EN_CLR	R/W0C	0h	Interrupt Enable Clear Register for msgmem_pend

22.2 Modular Controller Area Network (68xx MCAN)

22.2.1 MCAN Overview

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time control with a high level of security. CAN has high immunity to electrical interference and the ability to self-diagnose and repair data errors. In a CAN network, many short messages are broadcast to the entire network, which provides for data consistency in every node of the system.

The MCAN module supports both classic CAN and CAN FD (CAN with Flexible Data-Rate) specifications. CAN FD feature allows high throughput and increased payload per data frame. The classic CAN and CAN FD devices can coexist on the same network without any conflict.

The device supports one MCAN module connecting to the CAN network through external (for the device) transceiver for connection to the physical layer. The MCAN module supports up to 10 Mbit/s data rate and is compliant to the CAN protocol specification 2.0 part A, B and ISO 11898-1.

22.2.1.1 Features

The MCAN module implements the following features:

- Conforms with CAN Protocol 2.0 A, B and ISO 11898-1
- Full CAN FD support (up to 64 data bytes)
- AUTOSAR and SAE J1939 support
- Up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO, up to 32 elements
- Configurable Transmit Queue, up to 32 elements
- Configurable Transmit Event FIFO, up to 32 elements
- Up to 64 dedicated Receive Buffers
- Two configurable Receive FIFOs, up to 64 elements each
- Up to 128 filter elements
- Internal Loopback mode for self-test
- Maskable interrupts, two interrupt lines
- Two clock domains (CAN clock / Host clock)
- Parity / ECC support - Message RAM single error correction and double error detection (SECDED) mechanism
- Local power-down and wakeup support
- Timestamp Counter

Not supported features:

- Full Message Memory capacity (4352 words). Only 1600 words implemented.
- Debug on CAN (Debug DMA)

- Host bus read and write bursts
- Host bus firewall
- GPIO mode
- External (IO) Loopback mode
- Device clock domains monitoring (using DCC module)

22.2.2 MCAN Functional Description

The MCAN module performs CAN protocol communication according to the CAN protocol Specification 2.0 part A, B and ISO 11898-1. The bit rate can be programmed to values up to 10 Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

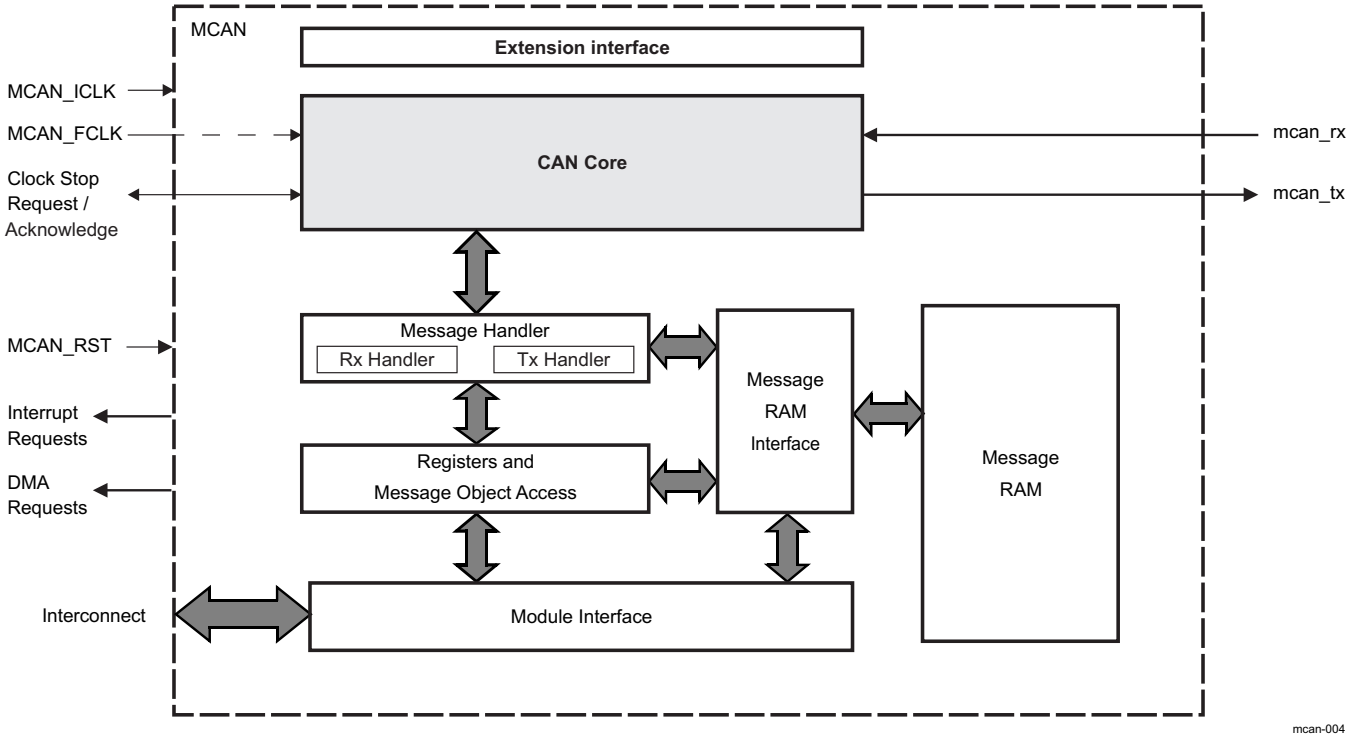
For communication on a CAN network, individual message frames can be configured. The message frames and identifier masks are stored in the Message RAM.

All functions concerning the handling of messages are implemented in the Message Handler.

The register set of the MCAN module can be accessed directly via the module interface. These registers are used to control and configure the CAN core and the Message Handler, and to access the Message RAM.

Figure 22-112 shows the MCAN module block diagram.

Figure 22-112. MCAN Block Diagram



The MCAN module blocks description:

- **CAN Core**: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1 protocol functions and supports 11-bit and 29-bit identifiers.
- **Message Handler**: the Message Handler (Rx Handler and Tx Handler) is a state machine that controls the data transfer between the single-ported Message RAM and the CAN core's Rx/Tx shift register. It also handles the acceptance filtering and the Interrupt/DMA request generation as programmed in the control registers.
- **Message RAM**: the main purpose of the Message RAM is to store Rx/Tx messages, Tx Event elements, and Message ID Filter elements (for more information, see Section 22.2.2.11, *Message RAM*).
- **Message RAM Interface**: enables connection between the Message RAM and the other blocks in the MCAN module.
- **Registers and Message Object Access**: Data consistency is ensured by indirect accesses to the message objects. During normal operation, all software and DMA accesses to the Message RAM are done through interface registers. The interface registers have the same word-length as the Message RAM.

- **Module Interface**: The MCAN module registers are accessed by the user software through a 32-bit peripheral bus interface.
- **Clocking**: Two clocks are provided to the MCAN module: the peripheral synchronous clock (interface clock - MCAN_ICLK) and the peripheral asynchronous clock (functional clock - MCAN_FCLK).
- **Extension Interface**: All flags from the Interrupt Register (MCAN_IR) as well as selected internal status and control signals are routed to this interface.

22.2.2.1 Module Clocking Requirements

Two clocks are provided to the MCAN module:

- the peripheral synchronous clock (MCAN_ICLK) as the general module clock source
- and the peripheral asynchronous clock (MCAN_FCLK) provided to the CAN core for generating the CAN bit timing.

Within the MCAN module there is a synchronization mechanism implemented to ensure safe data transfer between the two clock domains. There are synchronization between the signals from the Host clock domain to the CAN clock domain and vice versa and between the reset signal (MCAN_RST) to the Host clock domain and to the CAN clock domain.

NOTE: MCAN_ICLK must always be higher or equal to MCAN_FCLK, in order to achieve a stable functionality of the MCAN module. Here, also the frequency shift of the modulated MCAN_ICLK has to be considered:

$f_{0,\ ICLK}(OCP) \pm \Delta f_{FM,\ ICLK}(OCP) \geq f_{FCLK}$

For more information on how to configure the relevant clock source registers, see , *PRCM* and the device data manual.

22.2.2.2 Interrupt and DMA Requests

The MCAN module provides interrupt and DMA requests. They are configured via the Host CPU. The Suspend Mode prevents the interrupt and DMA requests from propagating to the Host CPU (for more information, see Section 22.2.2.4.8.2, *Suspend Mode*).

22.2.2.2.1 Interrupt Requests

The MCAN module has two interrupt lines. The first interrupt line (INT0) is associated with the MCAN core. There are 30 internal interrupt sources. The interrupts are 'level high' interrupts.

For more information, see the following registers:

- Interrupt Register (MCAN_IR)
- Interrupt Enable (MCAN_IE)
- Interrupt Line Select (MCAN_ILS)
- Interrupt Line Enable (MCAN_ILE)

The MCAN module is capable of issuing an ECC interrupt. After clearing the ECC interrupt source, the application software must also write '1' to MCANSS_ECC_EOI[8] ECC_EOI bit (for more information, see Section 22.2.2.7.2, *ECC Aggregator*).

The second interrupt line (INT1) is associated with the External Timestamp Counter. When the External Timestamp Counter rolls over it produces an interrupt (see Section 22.2.2.5.1, *External Timestamp Counter*).

For more information, see the following registers:

- Interrupt Clear Shadow Register (MCANSS_ICS)
- Interrupt Raw Status Register (MCANSS_IRS)
- Interrupt Enable Clear Shadow Register (MCANSS_IECS)
- Interrupt Enable Register (MCANSS_IE)

- Interrupt Enable Status (MCANSS_IES)
- End Of Interrupt (MCANSS_EOI)
- External Timestamp Prescaler (MCANSS_EXT_TS_PRESCALER)
- External Timestamp Unserviced Interrupts Counter (MCANSS_EXT_TS_UNSERVICED_INTR_CNTR)

22.2.2.2.2 DMA Requests

Functional transmit and Filter DMA requests are generated by the MCAN module based on the signaling in the Extention Interface. The DMA signaling uses a simple DMA request active high pulse. Only one Tx DMA event is provided by the MCAN module.

Standard and Extended message filters can be set to issue a pulse when a filter match occurs. These 'Filter Events' can be used to DMA messages from the Rx FIFO. The events are high level single clock cycle (MCAN_ICLK) pulses. Only two Filter DMA events are provided by the MCAN module.

22.2.2.3 Fuseable CAN FD Operation Enable

The Flexible Datarate feature of the MCAN module can be enabled by writing '1' to MCAN_CCCR[8] FDOE bit. A value of '0' on the primary configuration port (mcanss_enable_fdoe) will force the MCAN_CCCR[8] FDOE bit during write to the MCAN_CCCR register which will prevent the device from enabling and using the CAN FD mode.

22.2.2.4 Operating Modes

22.2.2.4.1 Software Initialization

Setting the MCAN_CCCR[0] INIT bit to '1' starts a software initialization. This is done either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off state. While the MCAN_CCCR[0] INIT bit is set, the message transfer is stopped and the status of the output mcan_tx pin is recessive (high). The counters of the Error Management Logic (EML) are unchanged. Setting the MCAN_CCCR[0] INIT bit does not change any configuration register. Resetting the MCAN_CCCR[0] INIT bit finishes the software initialization. After waiting for the occurrence of a sequence of 11 consecutive recessive bits (indication for Bus_Idle state) the message transfer starts.

Access to the MCAN configuration registers is only enabled when both MCAN_CCCR[0] INIT and MCAN_CCCR[1] CCE bits are set (write protection).

The MCAN_CCCR[1] CCE bit can only be set/reset while the MCAN_CCCR[0] INIT = '1'. The MCAN_CCCR[1] CCE bit is automatically reset when the MCAN_CCCR[0] INIT bit is reset.

The following registers are reset when the MCAN_CCCR[1] CCE bit is set:

- MCAN_HPMS - High Priority Message Status
- MCAN_RXF0S - Rx FIFO 0 Status
- MCAN_RXF1S - Rx FIFO 1 Status
- MCAN_TXFQS - Tx FIFO/Queue Status
- MCAN_TXBRP - Tx Buffer Request Pending
- MCAN_TXBTO - Tx Buffer Transmission Occurred
- MCAN_TXBCF - Tx Buffer Cancellation Finished
- MCAN_TXEFS - Tx Event FIFO Status

The Timeout Counter value MCAN_TOCV[15:0] TOC field is preset to the value configured by the MCAN_TOCC[31:16] TOP field when the MCAN_CCCR[1] CCE bit is set.

In addition the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR[1] CCE = '1'.

The following registers are only writeable while MCAN_CCCR[1] CCE = '0'

- MCAN_TXBAR - Tx Buffer Add Request
- MCAN_TXBCR - Tx Buffer Cancellation Request

MCAN_CCCR[7] TEST and MCAN_CCCR[5] MON bits can only be set by the Host CPU while MCAN_CCCR[0] INIT = '1' and MCAN_CCCR[1] CCE = '1'. Both bits may be reset at any time. The MCAN_CCCR[6] DAR bit can only be set/reset while MCAN_CCCR[0] INIT = '1' and MCAN_CCCR[1] CCE = '1'.

22.2.2.4.2 Normal Operation

Once the MCAN module is initialized and the MCAN_CCCR[0] INIT bit is reset to zero, the MCAN module synchronizes itself to the CAN bus and is ready for communication. After passing the acceptance filtering, received messages including Message Identifier (ID) and Data Length Code (DLC) are stored into a dedicated Rx Buffer or into Rx FIFO 0 / Rx FIFO 1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated.

NOTE: Automated transmission on reception of remote frames is not supported.

22.2.2.4.3 CAN FD Operation

There are two variants of CAN FD frame transmission:

- CAN FD frame transmission without bit rate switching
- CAN FD frame transmission where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame

In the CAN frames FDF = recessive (logical '1') signifies a CAN FD frame, FDF = dominant (logical '0') signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF - res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. Note that the coding of res = recessive is reserved for future expansion of the protocol. In case the MCAN module receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting the MCAN_PSR[14] EXE bit. When Protocol Exception Handling is enabled (MCAN_CCCR[12] PXHD = '0'), this causes the operation state to change from Receiver (MCAN_PSR[4:3] ACT = '10') to Integrating (MCAN_PSR[4:3] ACT = '00') at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR[12] PXHD = '1'), the MCAN will treat a recessive res bit as an form error and will respond with an error frame.

CAN FD operation is enabled by programming the MCAN_CCCR[8] FDOE bit. In case MCAN_CCCR[8] FDOE = '1', transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via the FDF bit in the respective Tx Buffer element.

With MCAN_CCCR[8] FDOE = '0', received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if the FDF bit of a Tx Buffer element is set. The MCAN_CCCR[8] FDOE and MCAN_CCCR[9] BRSE bits can only be changed while the MCAN_CCCR[0] INIT and MCAN_CCCR[1] CCE bits are both set. With MCAN_CCCR[8] FDOE = '0', the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format.

With MCAN_CCCR[8] FDOE = '1' and MCAN_CCCR[9] BRSE = '0', only FDF bit of a Tx Buffer element is evaluated. With MCAN_CCCR[8] FDOE = '1' and MCAN_CCCR[9] BRSE = '1', transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN

communication.

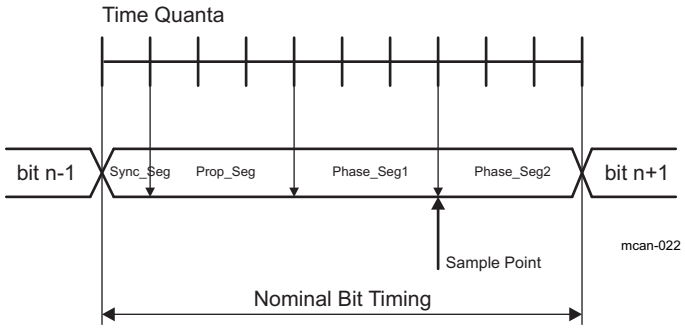
The coding of the DLC in the CAN FD format differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN (0 to 8 data bytes), the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to Table 22-108.

Table 22-108. DLC Coding in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

For CAN FD frames with bit rate switchingIn, the bit timing will be switched inside the frame after the BRS (Bit Rate Switch) bit in case this bit is recessive. In the CAN FD arbitration phase, before the BRS bit, the nominal CAN bit timing (see Figure 22-113) is used as configured by the Nominal Bit Timing and Prescaler Register MCAN_NBTP. In the following CAN FD data phase, the data phase bit timing is used as configured by the Data Bit Timing and Prescaler Register MCAN_DBTP. The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

Figure 22-113. CAN Bit Timing



The maximum configurable data phase bit timing depends on the CAN clock frequency (MCAN_FCLK). Example: with MCAN_FCLK = 20 MHz and the shortest configurable bit time of 4 t_q (time quanta), the bit rate in the data phase is 5 Mbit/s.

For both CAN FD without and CAN FD with bit rate switching the value of the ESI (Error Status Indicator) bit depends on transmitter's error state (see MCAN_PSR[11] RESI bit) monitored at the start of the transmission. If the transmitter has error passive flag the ESI bit is transmitted recessive, else it is transmitted dominant.

22.2.2.4.4 Transmitter Delay Compensation

22.2.2.4.4.1 Description

When only one CAN FD node is transmitting and all others are receivers the length of the bus line has no impact. When transmitting via the mcan_tx pin the MCAN module receives the transmitted data from its local CAN transceiver via the mcan_rx pin. The received data is delayed. If the transmitter delay is greater than TSEG1 (time segment before sample point), a bit error is detected.

The MCAN module provides a delay compensation mechanism to compensate the transmitter delay. The compensation mechanism enables transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Without transmitter delay compensation the bit rate in the data phase is limited by the transmitter delay.

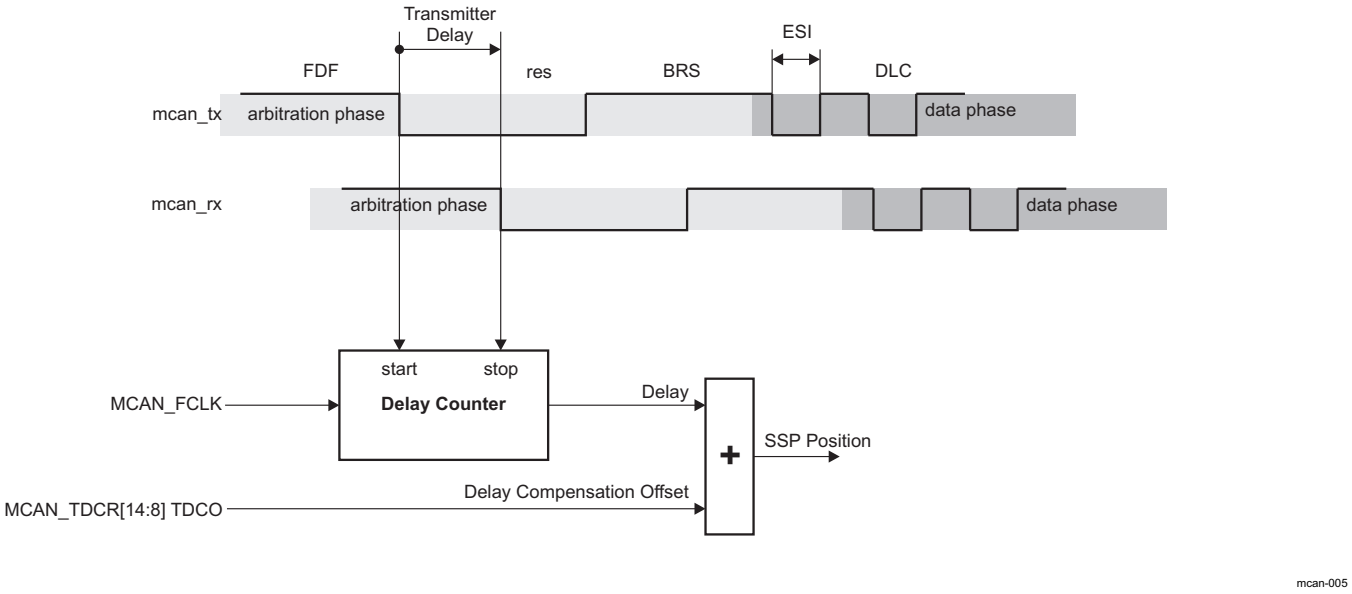
The mechanism enables configurations where the data bit time is shorter than the transmitter delay (it is described in detail in the new ISO11898-1). The transmitter delay compensation is enabled by setting the MCAN_DBTP[23] TDC bit to '1'.

The delayed transmit data is compared against the received data at the Secondary Sample Point (SSP) in order to check for bit errors during the data phase of transmitting nodes. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN's transmit output mcan_tx pin through the transceiver to the receive input mcan_rx pin plus the transmitter delay compensation offset configured by the MCAN_TDCR[14:8] TDCO field (see Figure 22-114). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (example: half of the bit time in the data phase). The position of the SSP is rounded down to the next integer number of mtq.

The actual transmitter delay compensation value can be checked by reading the MCAN_PSR[22:16] TDCV field. This filed is cleared when the MCAN_CCCR[0] INIT bit is set and is updated at each transmission of CAN FD frame while the MCAN_DBTP[23] TDC bit is set.

Figure 22-114. Transmitter Delay Measurement



22.2.2.4.4.2 Transmitter Delay Compensation Measurement

When transmitter delay compensation is enabled (by programming MCAN_DBTP[23] TDC = '1'), the measurement is started within each transmitted CAN FD frame at the falling edge of FDF bit to bit res. The measurement is stopped when this edge is seen at the receive input mcan_rx pin of the transmitter. The resolution of this measurement is one mtq (see Figure 22-114). The mtq (minimum time quantum) dimension is equal to the CAN clock period (MCAN_FCLK).

The use of a transmitter delay compensation filter window can be enabled by programming MCAN_TDCR[6:0] TDCF field. This filter feature defines a minimum value for the SSP position to avoid the case in which a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in an early taken SSP position. Dominant edges on the mcan_rx pin, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR[6:0] TDCF field and the mcan_rx pin is low.

The following boundary conditions have to be considered:

- The sum of the measured delay from the mcan_tx pin to the mcan_rx pin and the configured transmitter delay compensation offset (MCAN_TDCR[14:8] TDCO field) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from the mcan_tx pin to the mcan_rx pin and the configured transmitter delay compensation offset (MCAN_TDCR[14:8] TDCO) field has to be less or equal 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

22.2.2.4.5 Restricted Operation Mode

In Restricted Operation Mode the CAN node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The recieve and transmit error counters (MCAN_ECR[14:8] REC and MCAN_ECR[7:0] TEC) are frozen while CAN error logging (MCAN_ECR[23:16] CEL) is active. The Host CPU can set the MCAN module into Restricted Operation Mode by setting MCAN_CCCR[2] ASM bit. The bit can only be set by the Host CPU at any time when both MCAN_CCCR[2] CCE and MCAN_CCCR[1] INIT bits are set to '1'.

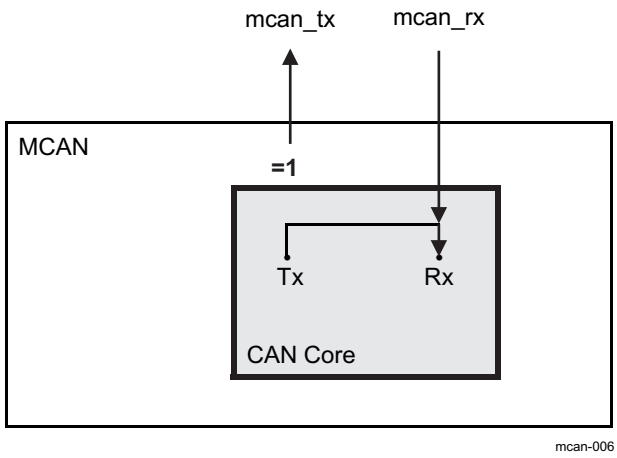
The Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the Host CPU has to reset MCAN_CCCR[2] ASM bit. This mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

NOTE: The Restricted Operation Mode must not be combined with the Loop Back Mode.

22.2.2.4.6 Bus Monitoring Mode

Entering Bus Monitoring Mode is done by setting the MCAN_CCCR[5] MON bit to '1'. In this mode (see ISO11898-1, *Bus Monitoring*), the MCAN module is able to receive valid data and remote frames, but cannot start a transmission. The MCAN module sends only recessive bits on the CAN bus. If the MCAN module is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN module monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode the MCAN_TXBRP register is held in reset state. The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 22-115 shows the connection of the mcan_tx and mcan_rx signals to the MCAN module in Bus Monitoring Mode.

Figure 22-115. Connection of Signals in Bus Monitoring Mode



22.2.2.4.7 Disabled Automatic Retransmission (DAR) Mode

According to the CAN Specification (see ISO11898-1, *Recovery Management*), the MCAN module provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled (see the MCAN_CCCR[6] DAR bit).

22.2.2.4.7.1 Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending MCAN_TXBRP[xx] TRPx bit is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

Successful transmission:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is not set

Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is set

Arbitration lost or frame transmission disturbed:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is not set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = '10' (transmission in spite of cancellation).

22.2.2.4.8 Power Down (Sleep Mode)

Entering Power Down mode is controlled via the input clock stop request signal (mcanss_clkstp_clkstop_req) or MCAN_CCCR[4] CSR bit. As long as the clock stop request signal is active, the MCAN_CCCR[4] CSR bit is read as '1'. When all pending transmission requests have completed, the MCAN module waits until bus idle state is detected. Then the MCAN module sets the MCAN_CCCR[1] INIT to '1' to prevent any further CAN transfers. The MCAN module acknowledges that it is ready for power down by setting the output clock stop acknowledge signal (mcanss_clkstop_clkstop_ack) to '1' and the MCAN_CCCR[3] CSA bit to '1'. In this state, before the clocks are switched off, further register accesses can be made. A write access to the MCAN_CCCR[1] INIT bit will have no effect. Now the module clock inputs MCAN_ICLK and MCAN_FCLK may be switched off.

To leave power down mode, the application has to turn on the module clocks before resetting the input clock stop request signal respectively the MCAN_CCCR[4] CSR flag bit. The MCAN will acknowledge this by resetting the output clock stop acknowledge signal respectively the MCAN_CCCR[3] CSA flag bit. Afterwards, the application can restart CAN communication by resetting MCAN_CCCR[1] INIT bit.

22.2.2.4.8.1 External Clock Stop Mode

The MCAN module supports two external clock stop modes:

- Immediate
- Graceful

In a graceful clock stop mode, when the clock stop request is asserted, the MCAN core will respond with clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. The MCAN_CCCR[0] INIT bit will be set, the MCAN core will go and stay Idle.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits to '1' (for more information, see Section 22.2.2.4.8.3, *Wakeup request*). When external clock stop request is removed and no suspend request is active, a read-modify-write to the MCAN_CCCR[0] INIT bit is performed to clear it.

22.2.2.4.8.2 Suspend Mode

The MCAN module supports two suspend modes:

- Immediate
- Graceful

In a graceful suspend mode (see the MCANSS_CTRL[3] FREE and MCANSS_CTRL[2] SOFT bits), when the suspend request is asserted, a clock stop request to the MCAN core is performed. The MCAN core will respond with clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. At that point the MCAN_CCCR[0] INIT bit will be set, the MCAN core will go and stay Idle. The suspend state can be verified by reading MCAN_CCCR[0] INIT bit.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits to '1' (for more information, see Section 22.2.2.4.8.3, *Wakeup request*). When suspend request is removed, if no external clock stop request is active, a read-modify-write to the MCAN_CCCR[0] INIT bit is performed to clear it.

During suspend mode the auto-clear feature is disabled. The following register fields have an auto-clear feature:

- MCAN_ECR[23:16] CEL
- MCAN_PSR[2:0] LEC
- MCAN_PSR[10:8] DLEC
- MCAN_PSR[11] RESI
- MCAN_PSR[12] RBRS
- MCAN_PSR[13] RFDF
- MCAN_PSR[14] PXE

22.2.2.4.8.3 Wakeup Request

Issuing a clock stop request puts the MCAN module into Power Down mode (Sleep Mode). During transition from IDLE to ACTIVE, if the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits are enabled, after the MCAN Core respond to the removal of the clock stop request with removing the clock stop acknowledge, a read-modify-write will be issued to clear the MCAN_CCCR[0] INIT bit and the MCAN core will resume operation.

If the MCANSS_CTRL[4] WAKEUPREQEN bit is set, the MCAN module provides a wakeup request (SWakeup) on any of the following wakeup events:

- The receive mcan_rx pin is dominant (logical 0)
- OCP access is performed

To clear the SWakeup in case any of these events is active, the MCANSS_CTRL[4] WAKEUPREQEN bit should be cleared. The MCAN module adds a third wakeup event source - interrupt line 0 (INT0). In this case the SWakeup is cleared by clearing the interrupt source.

22.2.2.4.9 Test Modes

The MCAN_TEST register write access is enabled by setting the test mode enable MCAN_CCCR[7] TEST bit to '1'. The MCAN_TEST register allows the configuration of the test modes and test functions.

The CAN transmit mcan_tx pin has four output functions. One of those functions can be selected by programming the MCAN_TEST[6:5] TX filed. Additionally to its default function (the serial data output) it can drive the CAN Sample Point signal to monitor the MCAN's bit timing and it can drive constant dominant or recessive values.

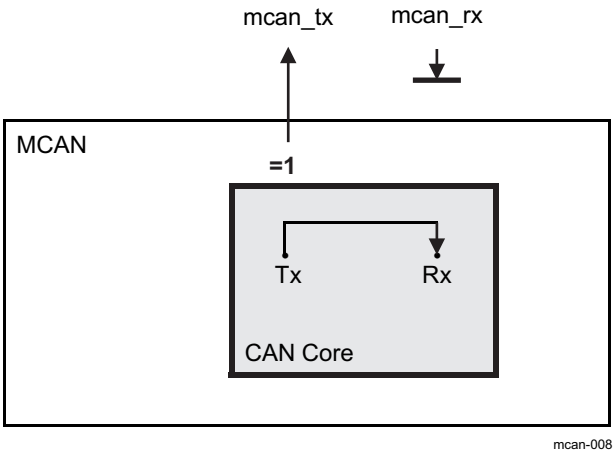
The actual value of the CAN receive mcan_rx pin can be monitored from MCAN_TEST[7] RX bit. Both functions can be used to check the CAN bus physical layer. Due to the synchronization mechanism between CAN clock (MCAN_FCLK) and Host clock (MCAN_ICLK) domain, there may be a delay of several Host clock periods between writing to the MCAN_TEST[6:5] TX filed until the new configuration is visible at the output mcan_tx pin. This applies also when reading input mcan_rx pin via the MCAN_TEST[7] RX bit.

NOTE: Test modes should be used for self test only. The software control for mcan_tx pin interferes with all CAN protocol functions. It is not recommended to use test modes for application.

22.2.2.4.9.1 Internal Loop Back Mode

The MCAN module can be set into Internal Loop Back Mode by programming MCAN_TEST[4] LBCK and MCAN_CCCR[5] MON bits to '1'. The Internal Loop Back Mode is used for a 'Hot Selftest'. The 'Hot Selftest' allows the MCAN module to be tested without affecting a running CAN system connected to the mcan_tx and mcan_rx pins. In this mode mcan_rx pin is disconnected from the MCAN module and mcan_tx pin is held recessive. Figure 22-116 shows the connection of the mcan_tx and mcan_rx pins to the MCAN module in case of Internal Loop Back Mode.

Figure 22-116. Internal Loop Back Mode



22.2.2.5 Timestamp Generation

The MCAN module has integrated a 16-bit wrap-around counter for timestamp generation. The timestamp counter prescaler MCAN_TSCC[19:16] TCP field can be configured to clock the counter in multiples of CAN bit times (1-16). The counter is readable via the MCAN_TSCV[15:0] TSC field. A write access to the MCAN_TSCV register resets the counter to zero. When the timestamp counter wraps around the interrupt MCAN_IR[16] TSW flag is set. On start of a frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element. For more information, see Section 22.2.2.11, *Message RAM*.

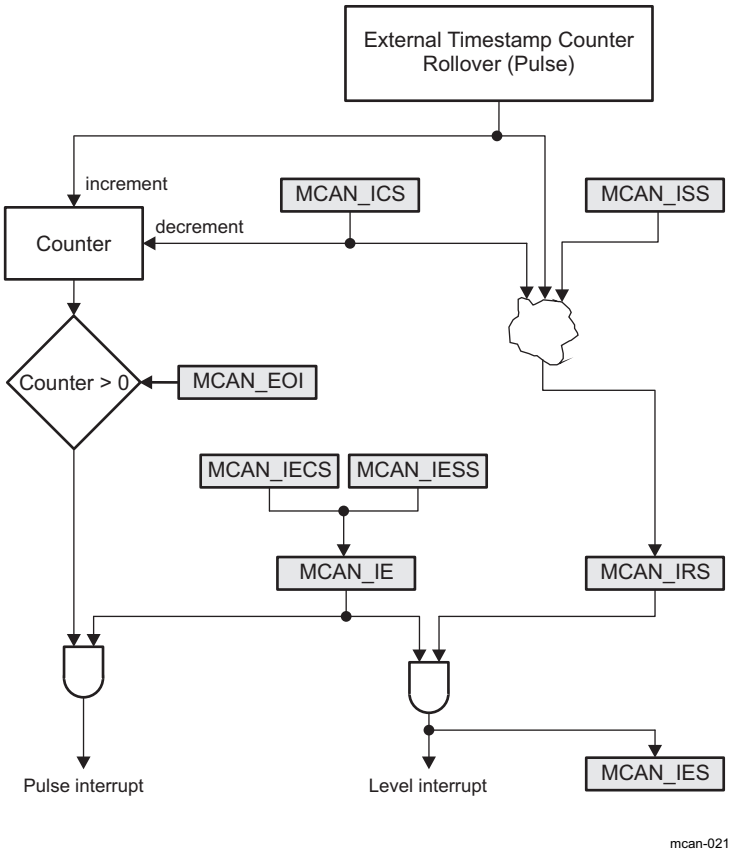
22.2.2.5.1 External Timestamp Counter

For CAN FD operation mode the MCAN core requires an External Timestamp Counter. An externally generated 16-bit vector may substitute the integrated 16-bit CAN bit time counter (internal timestamp counter) for receive and transmit timestamp generation. An external 16-bit timestamp counter can be used by programming the MCAN_TSCC[1:0] TSS field.

The External Timestamp Counter uses the interface clock (MCAN_ICLK) as a reference clock. The MCAN Core accepts a 16-bit timestamp. A 24-bit prescaler provides a programmable resolution for the timestamp (see MCANSS_EXT_TS_PRESCALER[23:0] PRESCALER field). When disabled the counter is reset back to zero. While enabled the counter keeps incrementing. When the timestamp rolls over the MCAN_IRQ_TS interrupt is generated. The MCAN module provides both pulse and level interrupt type for this interrupt.

When the timestamp rolls over the MCANSS_IRS register is set (see Figure 22-117). The MCANSS_IE register can be affected by writing to the MCAN_IESS register to set or to the MCANSS_IECS register to clear. The level interrupt is a reflection of both MCANSS_IRS and MCANSS_IE being set. The MCANSS_IES register reflects the level interrupt. When an rollover event occurs the interrupt counter is incremented. Writing to the MCANSS_ICS register to clear the MCANSS_IRS register will also decrement the interrupt counter. Writing to the MCANSS_EOI register will issue another pulse if the interrupt counter is not zero.

Figure 22-117. External Timestamp Counter Interrupt



22.2.2.6 Timeout Counter

The MCAN module has integrated a 16-bit Timeout Counter. It is used to signal timeout conditions for the Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO Message RAM elements. The Timeout Counter is configured via the MCAN_TOCC register. It is enabled via the MCAN_TOCC[0] ETOC bit. The Timeout Counter operates as down-counter and uses the same prescaler programmed by the MCAN_TSCC[19:16] TCP field as the Timestamp Counter. The actual counter value can be monitored from the MCAN_TOCV[15:0] TOC field. The Timeout Counter can be started only when MCAN_CCCR[1] INIT = '0' and stopped when MCAN_CCCR[1] INIT = '1' (example: when the MCAN enters Bus_Off state). The operation mode is selected by the MCAN_TOCC[2:1] TOS field. When Continuous Mode is selected, the counter starts when MCAN_TOCV[1] INIT = '0', a write to the MCAN_TOCV register presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field and continues down-counting.

In case the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field. Down-counting is started when the first FIFO element is stored. Writing to the MCAN_TOCV register has no effect. When the counter reaches zero, the interrupt MCAN_IR[18] TOO flag is set.

In Continuous Mode, the counter is immediately restarted at the value configured by the MCAN_TOCC[31:16] TOP field.

22.2.2.7 Safety

The Message Memory is wrapped in an ECC wrapper provididing SECDED parity functionalilty. The ECC wrapper is controlled by an ECC Aggregator.

22.2.2.7.1 ECC Wrapper

The ECC wrapper provides Single Error Correction (SEC) and Double Error Detection (DED) parity to the Message Memory content. It has side band signals for error notification. The ECC Wrapper implements an error injection test mode.

The error correction is done using a lazy write back. When an error is detected, it is noted in a FIFO Queue which waits for an access gap to write the data back and refresh the memory. If a transaction writes new data to the compromised entry before the lazy write back completes, the write back is discarded.

22.2.2.7.2 ECC Aggregator

This section describes the functional details of the ECC Aggregator module.

22.2.2.7.2.1 ECC Aggregator Overview

The ECC Aggregator module supports the following general features:

- Provides a mechanism to control and monitor the ECC RAM in the MCAN module.
- Provides software access to all the ECC related registers.
- Supports software readable status of ECC single/double-bit errors and associated info such as RAM address and data bit(s) that are in error.
- Aggregates level pending status from the ECC RAM into a single interrupt to the Host CPU.

The following feature is not supported:

- Statistics such as tracking the number of single and double-bit errors. If needed, these operations can be handled by software.

22.2.2.7.2.2 ECC Aggregator Registers

There are 3 groups of registers in the ECC aggregator module:

- Global registers - Aggregator Revision Register (MCANSS_ECC_AGGR_REVISION), ECC Vector Register (MCANSS_ECC_VECTOR), Misc Status Register (MCANSS_ECC_MISC_STATUS), ECC Control Register (MCANSS_ECC_CONTROL), and ECC Wrapper Revision Register (MCANSS_ECC_WRAP_REVISION).
- Control and status registers - ECC Error Control Registers (MCANSS_ECC_ERR_CTRL1 and MCANSS_ECC_ERR_CTRL2) and ECC Error Status Registers (MCANSS_ECC_ERR_STAT1 and MCANSS_ECC_ERR_STAT2).
- Interrupt registers - interrupt status, interrupt enable set, interrupt enable clear and EOI (End Of Interrupt) registers that are part of a standard interrupt module. For more information, see the following registers:
  - MCANSS_ECC_SEC_EOI_REG
  - MCANSS_ECC_SEC_STATUS_REG0
  - MCANSS_ECC_SEC_ENABLE_SET_REG0
  - MCANSS_ECC_SEC_ENABLE_CLR_REG0
  - MCANSS_ECC_DED_EOI_REG
  - MCANSS_ECC_DED_STATUS_REG0
  - MCANSS_ECC_DED_ENABLE_SET_REG0
  - MCANSS_ECC_DED_ENABLE_CLR_REG0

22.2.2.7.2.3 Reads to ECC Control and Status Registers

The reads to the ECC control and status registers are triggered by writing a 'read message' to the ECC Vector Register as described below:

- Software writes value (the ECC RAM ID) to the MCANSS_ECC_VECTOR[10-0] ECC_VECTOR field to select the ECC RAM for control or status.

- Software writes '1' to the MCANSS_ECC_VECTOR[15] RD_SVBUS bit to trigger a read.
- Software writes read address to the MCANSS_ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field.
- Software then polls the MCANSS_ECC_VECTOR[24] RD_SVBUS_DONE bit to check if it is '1'. This bit indicates that the read operation has completed.
- Software reads the data from the ECC control or status register. The following clock cycle (MCAN_ICLK) returns the read data.

22.2.2.7.2.4 ECC Interrupts

The ECC aggregator module aggregates the level pending status from the ECC RAM into a single EOI-handshake based interrupt to the Host CPU. Software is expected to follow the sequence described below:

- Software enables the interrupts for the ECC RAM by writing to the MCANSS_ECC_SEC_ENABLE_SET_REG0 / MCANSS_ECC_DED_ENABLE_SET_REG0 register.
- Software writes the ECC RAM ID in the MCANSS_ECC_VECTOR[10-0] ECC_VECTOR.
- Software writes the MCANSS_ECC_VECTOR[15] RD_SVBUS bit to trigger the read.
- Software writes the MCANSS_ECC_ERR_STAT1 register address to the MCANSS_ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field. Software will need to load the 'read message' in the MCANSS_ECC_VECTOR register again if it needs to read the MCANSS_ECC_ERR_STAT2 register.
- Software polls the MCANSS_ECC_VECTOR[24] RD_SVBUS_DONE bit. When this bit is set, a read of the MCANSS_ECC_ERR_STAT1 / MCANSS_ECC_ERR_STAT2 register is performed.
- After the interrupt has been serviced, software will clear the interrupt status by writing to the MCANSS_ECC_ERR_STAT1[8] CLR_ECC_SEC or MCANSS_ECC_ERR_STAT1[9] CLR_ECC_DED bit depending on the type of the ECC error.
- Software has to poll the MCANSS_ECC_ERR_STAT1 register to guarantee that the status bit has been cleared.
- Software will write to the MCANSS_ECC_SEC_EOI_REG / MCANSS_ECC_DED_EOI_REG register to clear the interrupt.
- After clearing the ECC interrupt source, the application software must also write '1' to the MCANSS_ECC_EOI[8] ECC_EOI bit.

22.2.2.8 Rx Handling

The Rx Handler controls the following operations:

- Acceptance filtering
- The transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1)
- Rx FIFO Put and Get Index operations

22.2.2.8.1 Acceptance Filtering

The MCAN module is capable to configure two sets of acceptance filters - one set for standard and one set for extended identifiers. These filters can be assigned to an Rx Buffer or to one of the two Rx FIFOs.

The main features of the filter elements are:

- Each filter element can be configured as:
  - Range Filter (from - to)
  - Filter for specific IDs (for one or two dedicated IDs)
  - Classic Bit Mask Filter
- Each filter element can be enabled / disabled individually
- Each filter element can be configured for acceptance or rejection filtering
- Filters are checked sequentially and execution (acceptance filtering procedure) stops at the first matching filter element or when the end of the filter list is reached

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register
- Extended ID AND Mask (MCAN_XIDAM) register

Depending on the configuration of the filter element (see SFEC/EFEC in Section 22.2.2.11, *Message RAM*) if filter matches, one of the following actions is performed:

- Received frame is stored in FIFO 0 or FIFO 1
- Received frame is stored in Rx Buffer
- Received frame is stored in Rx Buffer and generation of pulse at filter event pin is performed. This is high level single MCAN_ICLK pulse. For more information, see Section 22.2.2.2.1, *DMA Requests*.
- Received frame is rejected
- Set High Priority Message interrupt flag MCAN_IR[8] HPM
- Set High Priority Message interrupt flag MCAN_IR[8] HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering starts when complete Message ID is received. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If a filter element matches - the Rx Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If an error condition occurs (for example: CRC error), this message is rejected with the following impact on the affected Rx Buffer or Rx FIFO:

- Rx Buffer:
  New Data flag (MCAN_NDAT1 / MCAN_NDAT2) of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data (for error type see MCAN_PSR[2:0] LEC respectively MCAN_PSR[10:8] DLEC fields).
- Rx FIFO:
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data (for error type see MCAN_PSR[2:0] LEC respectively MCAN_PSR[10:8] DLEC fields). If matching Rx FIFO is configured to operate in overwrite mode, the boundary conditions described in Section 22.2.2.8.2.2 have to be considered.

22.2.2.8.1.1 Range Filter

Each filter element can be configured to operate as Range Filter (Standard Filter Type SFT = '00' / Extended Filter Type EFT = '00'). The filter matches for all received message frames with IDs in the range from SFID1 to SFID2 (SFID2 ≥ SFID1) respectively in the range from EFID1 to EFID2 (EFID2 ≥ EFID1). For more information see Section 22.2.2.11.5, *Standard Message ID Filter Element* and Section 22.2.2.11.6, *Extended Message ID Filter Element*.

There are two options for range filtering of extended frames:

- Extended Filter Type EFT = '00': The Extended ID AND Mask (MCAN_XIDAM) is used for Range Filtering. The Message ID of received frames is ANDed with the Extended ID AND Mask (MCAN_XIDAM) before the range filter is applied.
- Extended Filter Type EFT = '11': The Extended ID AND Mask (MCAN_XIDAM) is not used for Range Filtering.

22.2.2.8.1.2 Filter for Specific IDs

Each filter element can be configured to filter one or two dedicated Message IDs (Standard Filter Type SFT ='01' / Extended Filter Type EFT ='01'). To filter only one specific Message ID, the filter element has to be configured with SFID1 = SFID2 respectively EFID1 = EFID2. For more information see Section 22.2.2.11.5, *Standard Message ID Filter Element* and Section 22.2.2.11.6, *Extended Message ID Filter Element*.

22.2.2.8.1.3 Classic Bit Mask Filter

Classic bit mask filtering can filter groups of Message IDs (Standard Filter Type SFT ='10' / Extended Filter Type EFT ='10'). This is done by masking single bits of a received Message ID. In this case SFID1/EFID1 element is used as Message ID filter, while SFID2/EFID2 element is used as filter mask.

A '0' bit at the filter mask (SFID2/EFID2) will mask out the corresponding bit position of the configured Message ID filter (SFID1/EFID1) and the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are '1' are relevant for acceptance filtering.

There are two interesting cases:

- All mask bits are '1': a match occurs only when the received Message ID and the configured Message ID filter are identical.
- All mask bits are '0': all Message IDs match.

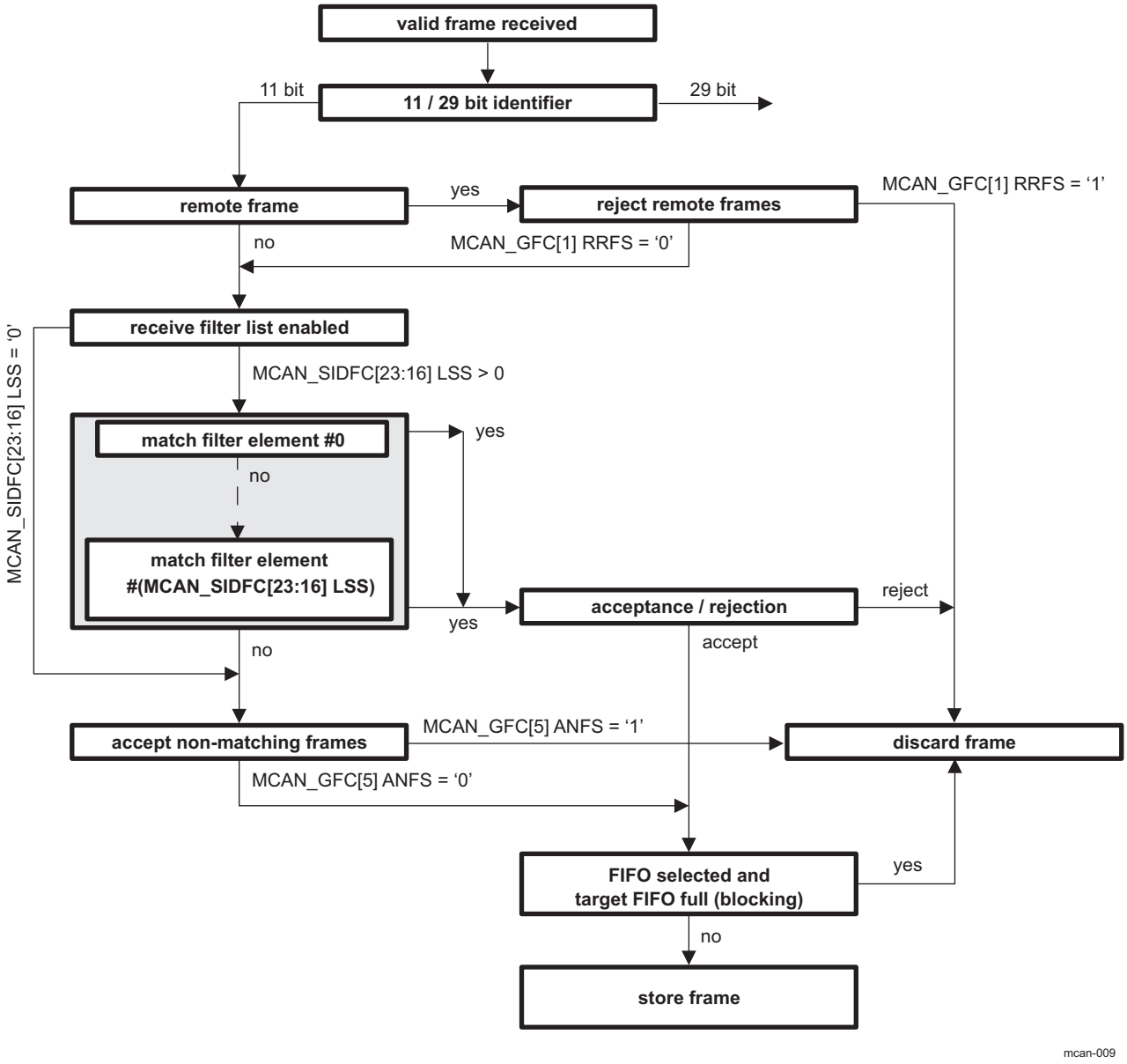
22.2.2.8.1.4 Standard Message ID Filtering

The standard Message ID (11-bit ID) filtering flow is shown in Figure 22-118. Section 22.2.2.11.5, *Standard Message ID Filter Element* describes the standard Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register

Figure 22-118. Standard Message ID Filter Path



22.2.2.8.1.5 Extended Message ID Filtering

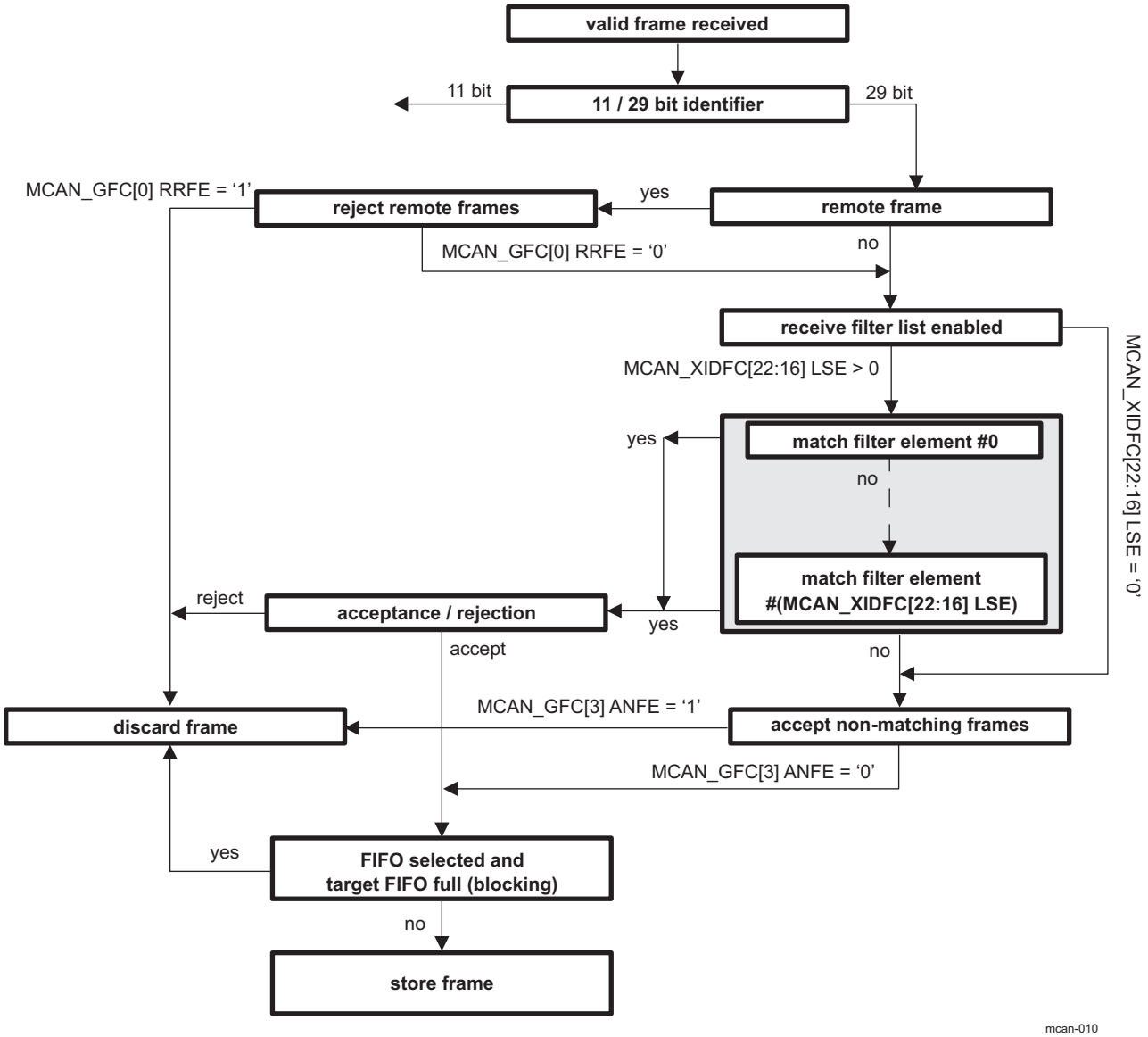
The extended Message ID (29-bit ID) filtering flow is shown in Figure 22-119. Section 22.2.2.11.6, *Extended Message ID Filter Element* describes the extended Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register

Note that before the filter list is executed the received identifier is ANDed with the Extended ID AND Mask (MCAN_XIDAM).

Figure 22-119. Extended Message ID Filter Path



22.2.2.8.2 Rx FIFOs

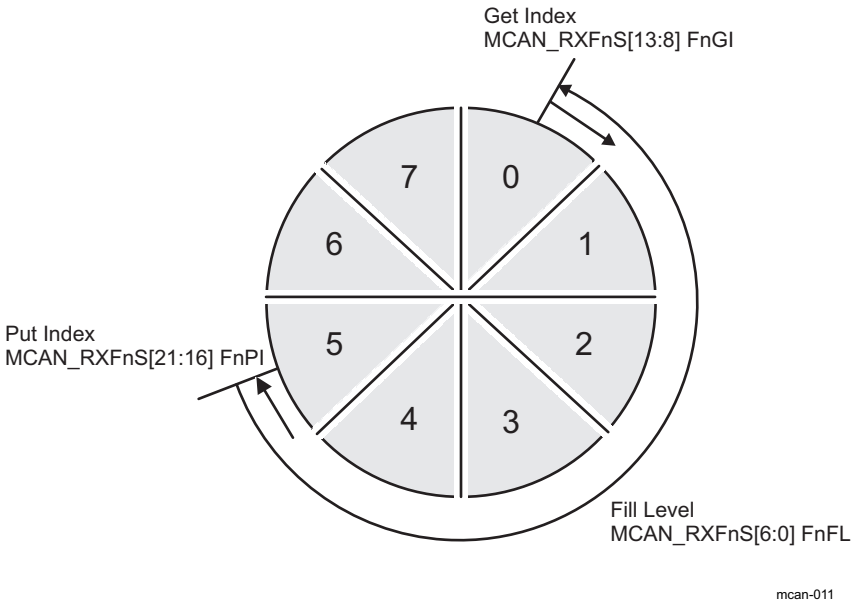
The configuration of the Rx FIFOs (Rx FIFO 0 and Rx FIFO 1) can be done via the MCAN_RXF0C and MCAN_RXF1C registers. Each Rx FIFO can be configured to store up to 64 received messages.

After acceptance filtering the received messages that passed are transferred to the Rx FIFO. The filter mechanisms available for the Rx FIFO 0 and Rx FIFO 1 is described in Section 22.2.2.8.1, *Acceptance Filtering*. Section 22.2.2.11.2, *Rx Buffer and FIFO Element* describes the Rx FIFO element.

The Rx FIFO watermark can be used to prevent an Rx FIFO overflow. If the Rx FIFO fill level reaches the Rx FIFO watermark configured by the MCAN_RXFnC[30:24] FnWM filed (where: n = 0 or 1) an interrupt flag MCAN_IR[1] RF0W / MCAN_IR[5] RF1W is set.

When the Rx FIFO Put Index reaches the Rx FIFO Get Index (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) an Rx FIFO Full condition is signalled by the MCAN_RXFnS[24] FnF status bit and interrupt flag MCAN_IR[2] RF0F / MCAN_IR[6] RF1F is set. Figure 22-120 shows Rx FIFO Status. The FIFOs fill level is presented in the MCAN_RXFnS[6:0] FnFL field (the number of elements stored in Rx FIFO).

Figure 22-120. Rx FIFO Status



Rx FIFOs start address in the Message RAM (MCAN_RXFnC[15:2]FnSA field) have to be configured when reading from an Rx FIFO (Rx FIFO Get Index - MCAN_RXFnS[13:8] FnGI). Table 22-109 presents Rx Buffer / Rx FIFO Element Size for different Rx Buffer / Rx FIFO Data Field Size which is configured via the MCAN_RXESC register.

Table 22-109. Rx Buffer / Rx FIFO Element Size

MCAN_RXESC[10:8] RBDS MCAN_RXESC[2:0] F0DS / MCAN_RXESC[6:4] F1DS	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

22.2.2.8.2.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is the default operation mode for the Rx FIFOs. It is configured by the MCAN_RXFnC[31] FnOM = '0'.

If an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by the MCAN_RXFnS[24] FnF = '1' and interrupt flag MCAN_IR[2] RF0F / MCAN_IR[6] RF1F is set.

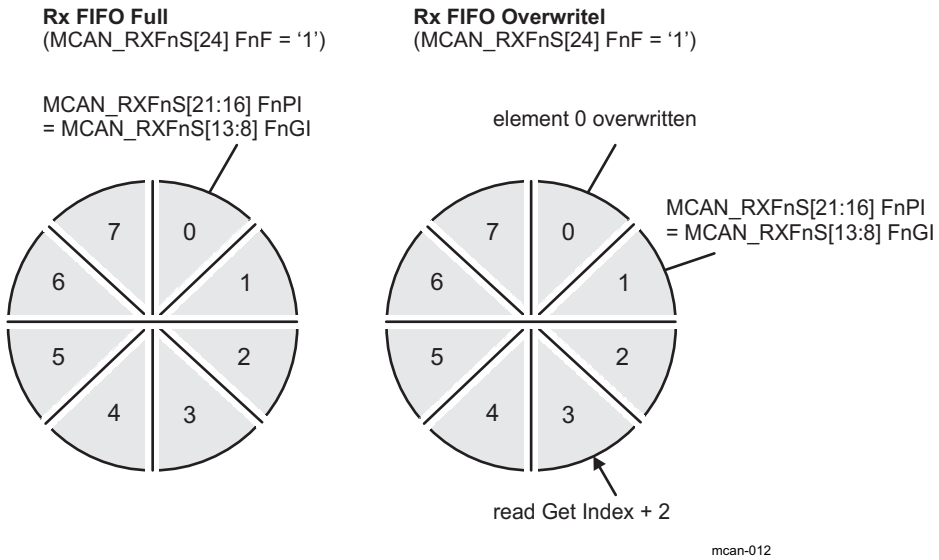
In case a message is received while the corresponding Rx FIFO is full, this message is rejected and the message lost condition is signalled by MCAN_RXFnS[25] RFnL = '1' and interrupt flag MCAN_IR[3] RFnL / MCAN_IR[25] RFnL is set.

22.2.2.8.2.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by the MCAN_RXFnC[31] FnOM = '1'. When an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) signalled by MCAN_RXFnS[24] FnF = '1', the next accepted message for the FIFO will overwrite the oldest FIFO message. Put index / Get index are both incremented by one.

In overwrite mode if an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (Put index) while the Host CPU is reading from the Message RAM (Get index). In this case inconsistent data may be read from the respective Rx FIFO element. The problem is solved by adding an offset to the Get index when reading from the Rx FIFO. The offset depends on how fast the Host CPU accesses the Rx FIFO. Figure 22-121 shows an offset of two with respect to the Get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 22-121. Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA[5:0] FnAI. This increments the get index to that element number. In case the Put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS[24] FnF = '0').

22.2.2.8.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx Buffers. The start address of the Rx Buffers section in the Message RAM is configured via MCAN_RXBC[15:2] RBSA field. To store in an Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = '111' and SFID2 / EFID2[10:9] = '00' has to be configured (see Section 22.2.2.11.5, *Standard Message ID Filter Element* and Section 22.2.2.11.6, *Extended Message ID Filter Element*).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element (the format is the same as for an Rx FIFO element). In addition the flag MCAN_IR[19] DRX (Message stored in Dedicated Rx Buffer) is set.

Table 22-110 shows Example Filter Configuration for Rx Buffers.

Table 22-110. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register MCAN_NDAT1 / MCAN_NDAT2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the Host CPU by writing a '1' to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

22.2.2.8.3.1 Rx Buffer Handling

Rx Buffer Handling include the following steps:

- Reset interrupt flag MCAN_IR[19] DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

22.2.2.9 Tx Handling

The Tx Handler is used to handle the Tx requests. It controls the transfer of transmit messages from the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue to the CAN Core, the Tx Event FIFO, and the Put and Get Index operations. The MCAN module supports up to 32 Tx Buffers. These Tx Buffers can be configured as dedicated Tx Buffers, Tx FIFO, or Tx Queue and as combination of dedicated Tx Buffers / Tx FIFO or dedicated Tx Buffers / Tx Queue. For each Tx Buffer element Classical CAN or CAN FD transmission mode can be configured. Section 22.2.2.11.3 describes the Tx Buffer Element. Table 22-111 shows the possible configurations for message transmission.

Table 22-111. Possible Configurations for Message Transmission

MCAN_CCCR		Tx Buffer Element		Frame Transmission
MCAN_CCCR[9] BRSE	MCAN_CCCR[8] FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	CAN FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	CAN FD without bit rate switching
1	1	1	1	CAN FD with bit rate switching

When the Tx Buffer Request Pending MCAN_TXBRP register is updated, or when a transmission has been started the Tx Handler starts scanning to check for the highest priority pending Tx request. The Tx Buffer with lowest Message ID has highest priority.

NOTE: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

22.2.2.9.1 Transmit Pause

The transmit pause feature is intended for use in CAN networks where the CAN Message IDs are specific and cannot easily be changed. These Message IDs may have a higher priority than other defined Message IDs, while in a specific application their relative priority should be inverse. This allows for a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed (paused).

The transmit pause feature is enabled by the MCAN_CCCR[14] TXP bit. By default this bit is disabled (MCAN_CCCR[14] TXP = '0'). Each time after successfully transmitted message, a pause for two CAN bit times occurs before the start of the next transmission. This allows the other CAN nodes in the network to transmit messages even if their Message IDs have lower priority.

22.2.2.9.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the Host CPU.

There are two options:

- Each dedicated Tx Buffer is configured with a specific Message ID.
- Two or more dedicated Tx Buffers are configured with the same Message ID. In this case the Tx Buffer with the lowest buffer number is transmitted first.

After the data section has been updated, a transmission is requested by an Add Request. This is done via the MCAN_TXBAR[x]ARn bit (where x = 0 - 31). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

Table 22-112 shows Tx Buffer / Tx FIFO / Tx Queue Element Size. A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM. The start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index from 0 to 31 (MCAN_TXFQS[20:16] TFQPI) * Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

Table 22-112. Tx Buffer / Tx FIFO / Tx Queue Element Size

MCAN_TXESC[2:0] TBDS	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

22.2.2.9.3 Tx FIFO

Tx FIFO mode is configured by setting bit MCAN_TXBC[30] TFQM = '0'. The stored in the Tx FIFO messages are transmitted starting with the message referenced by the Get Index MCAN_TXFQS[12:8] TFGI field. After each transmission the Get Index is incremented until the Tx FIFO is empty. The Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field indicates the number of the available free Tx FIFO elements. The Tx FIFO allows transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS[20:16] TFQPI field. After each Add Request (MCAN_TXBAR[x] ARn = '1') the Put Index is incemented to the next free Tx FIFO element. When the Put Index reaches the Get Index (MCAN_TXFQS[20:16] TFQPI = MCAN_TXFQS[12:8] TFGI), Tx FIFO Full condition is signalled by bit MCAN_TXFQS[21] TFQF = '1'. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field.

In case a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field is recalculated. In case transmission cancellation is applied to any other Tx Buffer - the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 22-112). The start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS[20:16] TFQPI (from 0 to 31) * Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

22.2.2.9.4 Tx Queue

Tx Queue mode is configured by setting bit MCAN_TXBC[30] TFQM = '1'. The stored in the Tx Queue messages are transmitted starting with the highest priority message (lowest Message ID). In case two or more Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS[20:16] TFQPI field. Each Add Request cyclically increments the Put Index to the next free Tx Buffer. In case of Tx Queue Full condition (MCAN_TXFQS[21] TFQF = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use the MCAN_TXBRP register instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 22-112). The start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS[20:16] TFQPI (from 0 to 31) * Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

22.2.2.9.5 Mixed Dedicated Tx Buffers / Tx FIFO

For this combination the Tx Buffers section in the Message RAM is separated in two parts:

- Dedicated Tx Buffers: the number of Dedicated Tx Buffers is configured by the MCAN_TXBC[21:16] NDTB field
- Tx FIFO: the number of Tx Buffers assigned to the Tx FIFO is configured by the MCAN_TXBC[29:24] TFQS field

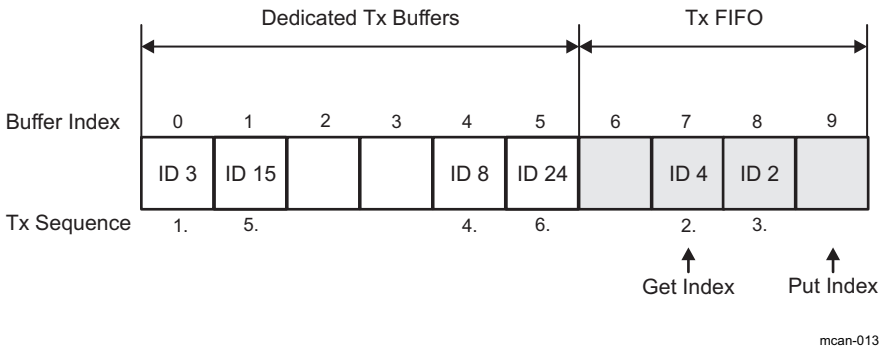
If the MCAN_TXBC[29:24] TFQS field is empty (zero) - only Dedicated Tx Buffers are used.

Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by the MCAN_TXFQS[12:8] TFGI field)
- Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 22-122 shows Mixed Dedicated Tx Buffers / Tx FIFO example.

Figure 22-122. Mixed Dedicated Tx Buffers / Tx FIFO (example)



22.2.2.9.6 Mixed Dedicated Tx Buffers / Tx Queue

For this combination the Tx Buffers section in the Message RAM is separated in two parts:

- Dedicated Tx Buffers: the number of Dedicated Tx Buffers is configured by the MCAN_TXBC[21:16] NDTB field
- Tx Queue: the number of Tx Buffers assigned to the Tx Queue is configured by the MCAN_TXBC[29:24] TFQS field

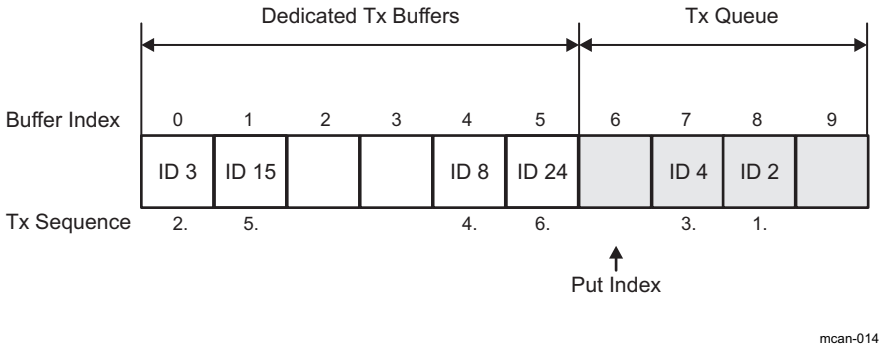
If MCAN_TXBC[29:24] TFQS field is empty (zero) - only Dedicated Tx Buffers are used.

Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 22-123 shows Mixed Dedicated Tx Buffers / Tx Queue example.

Figure 22-123. Mixed Dedicated Tx Buffers / Tx Queue (example)



22.2.2.9.7 Transmit Cancellation

This feature is especially intended for gateway and AUTOSAR based applications. The Host CPU can cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer by setting bit MCAN_TXBCR[n] CRn = '1' (where n = 0 - 31). The corresponding bit position n is equivalent to the number of the Tx Buffer.

Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of the MCAN_TXBCF register (MCAN_TXBCF[n] CFn = '1').

If transmission from a Tx Buffer is already ongoing and a transmit cancellation is requested, the corresponding MCAN_TXBRP[n] TRPn bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO[n] TOn and MCAN_TXBCF[n] CFn bits are set. If the transmission was not successful, only the corresponding bit MCAN_TXBCF[n] CFn = '1'.

NOTE: If pending transmission is cancelled immediately before this transmission could have been started, a short time window occurs where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

22.2.2.9.8 Tx Event Handling

To support Tx Event Handling the Message RAM has implemented a Tx Event FIFO section. Up to 32 Tx Event FIFO elements can be configured. Section 22.2.2.11.4 describes the Tx Event FIFO element. After message transmission on the CAN bus, Message ID and Timestamp are stored in a Tx Event FIFO element. To link a Tx Event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

A Tx Event FIFO full condition is signalled by the MCAN_IR[14] TEFF bit. In this case no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented (MCAN_TXEFS[12:8] EFGI). In case a Tx Event occurs while the Tx Event FIFO is full, this event is rejected and interrupt flag MCAN_IR[15] TEFL bit is set.

The Tx Event FIFO watermark can be configured to avoid a Tx Event FIFO overflow. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by the MCAN_TXEFC[29:24] EFWM field, interrupt flag MCAN_IR[13] TEFW is set. When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS[12:8] EFGI field has to be added to the Tx Event FIFO start address MCAN_TXEFC[15:2] EFSA field.

22.2.2.10 FIFO Acknowledge Handling

The Get Indices of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1) and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see MCAN_RXF0A, MCAN_RXF1A, and MCAN_TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level.

There are two use cases:

- A single element has been read from the FIFO: the Get Index value is written to the FIFO Acknowledge Index.
- A sequence of elements has been read from the FIFO: the Get Index value (Index of the last element read) is written to the FIFO Acknowledge Index at the end of that read sequence.

The Host CPU has free access to the Message RAM. The special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This can be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also changes the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

NOTE: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN module does not check for erroneous values.

22.2.2.11 Message RAM

The MCAN module has implemented Message RAM. The main purpose of the Message RAM is to store:

- Receive Messages
- Transmit Messages
- Tx Event Elements
- Message ID Filter Elements

22.2.2.11.1 Message RAM Configuration

The MCAN module is configured to allocate 4352 words in the Message RAM. The Message RAM has a width of 32 bits.

The address range of the Message RAM is from 0xFF50 0000 to 0xFF50 43FC.

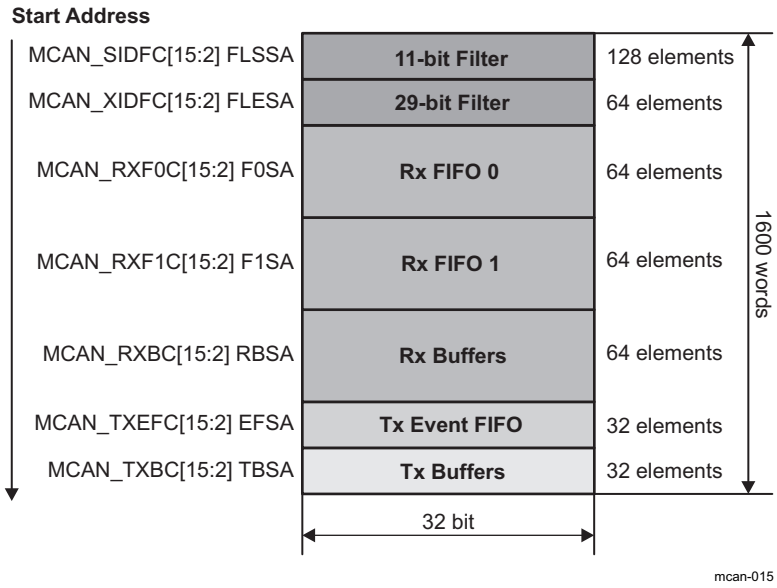
The Message RAM is capable to include each of the sections listed in Figure 22-124. It is not necessary to configure each of the sections (a section in the Message RAM may be 0) and there is not restriction with respect to the sequence of the sections. For parity checking or ECC a respective number of bits has to be added to each word.

When the MCAN module addresses the Message RAM it addresses 32-bit words. The start addresses are configurable and they are 32-bit word addresses.

The element size can be configured for:

- Rx FIFO 0 via the MCAN_RXESC[2:0] F0DS field
- Rx FIFO 1 via the MCAN_RXESC[6:4] F1DS field
- Rx Buffers via the MCAN_RXESC[10:8] RBDS field
- Tx Buffers via the MCAN_TXESC[2:0] TBDS field

Figure 22-124. Message RAM Configuration



The Host CPU configures the following information in the Message RAM:

- Start addresses of the memory sections
- Number of elements in each section
- The size of the elements in some sections

NOTE: The MCAN module does not check for errors in the Message RAM configuration. The configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully. This will prevent falsification or loss of data.

22.2.2.11.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via the MCAN_RXESC register.

Figure 22-125 shows Rx Buffer / Rx FIFO element structure.

Figure 22-125. Rx Buffer / Rx FIFO Element Structure

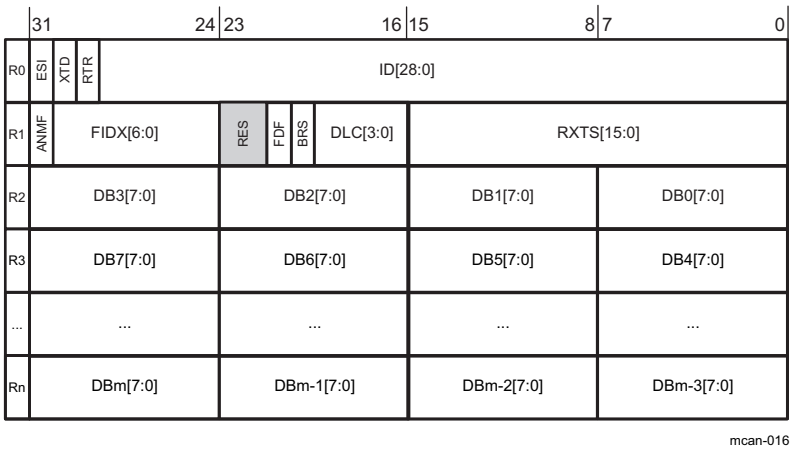


Table 22-113 shows Rx Buffer / Rx FIFO element field descriptions.

Table 22-113. Rx Buffer / Rx FIFO Element Field Descriptions

Word	Bits	Field Name	Description
R0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier Signals to the Host CPU whether the received frame has a standard or extended identifier. • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request Signals to the Host CPU whether the received frame is a data frame or a remote frame. • 0x0: Received frame is a data frame • 0x1: Received frame is a remote frame **Note:** There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = '1'), RTR bit reflects the state of the reserved r1 bit (RES[23]).
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier is stored into ID[28:18].
R1	31	ANMF	Accepted Non-matching Frame Acceptance of non-matching frames may be enabled via the MCAN_GFC[5:4] ANFS and MCAN_GFC[3:2] ANFE fields. • 0x0: Received frame matching filter index FIDX field • 0x1: Received frame did not match any Rx filter element
	30:24	FIDX[6:0]	Filter Index 0x0-0x7F (0-127): Index of matching Rx acceptance filter element (invalid if ANMF = '1'). Range is 0 to MCAN_SIDFC[23:16] LSS - 1 respectively MCAN_XIDFC[22:16] LSE - 1.
	23:22	RES	Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame received without bit rate switching • 0x1: Frame received with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: received frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: received frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: received frame has 12/16/20/24/32/48/64 data bytes
	15:0	RXTS[15:0]	Rx Timestamp Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC[19:16] TCP.
R2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0

Table 22-113. Rx Buffer / Rx FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
R3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Rn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3-17) are used for storage of a CAN message's data field.

22.2.2.11.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler makes difference between dedicated Tx Buffers and Tx FIFO / Tx Queue via the MCAN_TXBC[29:24] TFQS and MCAN_TXBC[21:16] NDTB fields. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via the MCAN_TXESC register.

Figure 22-126 shows Tx Buffer element structure.

Figure 22-126. Tx Buffer Element Structure

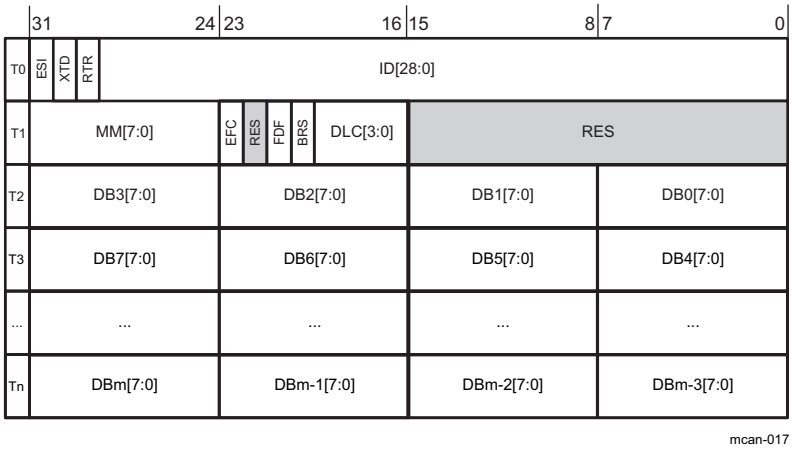


Table 22-114 shows Tx Buffer element field descriptions.

Table 22-114. Tx Buffer Element Field Descriptions

Word	Bits	Field Name	Description
T0	31	ESI	Error State Indicator • 0x0: ESI bit in CAN FD format depends only on error passive flag • 0x1: ESI bit in CAN FD format transmitted recessive **Note:** The ESI bit of the transmit buffer is or'ed with the error passive flag to decide the value of the ESI bit in the transmitted CAN FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Transmit data frame • 0x1: Transmit remote frame **Note:** When RTR = 1, the MCAN module transmits a remote frame according to ISO11898-1, even if the MCAN_CCCR[8] FDOE bit enables the transmission in CAN FD format.
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].
T1	31:24	MM[7:0]	Message Marker Written by Host CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 22-115).
	23	EFC	Event FIFO Control • 0x0: Don't store Tx events • 0x1: Store Tx events
	22	RES	Reserved
	21	FDF	FD Format • 0x0: Frame transmitted in Classic CAN format • 0x1: Frame transmitted in CAN FD format
	20	BRS	Bit Rate Switch • 0x0: CAN FD frames transmitted without bit rate switching • 0x1: CAN FD frames transmitted with bit rate switching **Note:** ESI, FDF, and BRS bits are only evaluated when CAN FD operation is enabled vie the MCAN_CCCR[8] FDOE bit. BRS bit is only evaluated when in addition the MCAN_CCCR[9] BRSE = '1'.
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: transmit frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: transmit frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes
	15:0	RES	Reserved
T2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0

Table 22-114. Tx Buffer Element Field Descriptions (continued)

Word	Bits	Field Name	Description
T3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Tn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3-17) are used for storage of a CAN message's data field.

22.2.2.11.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from the MCAN_TXEFS register.

Figure 22-127 shows Tx Event FIFO element structure.

Figure 22-127. Tx Event FIFO Element Structure

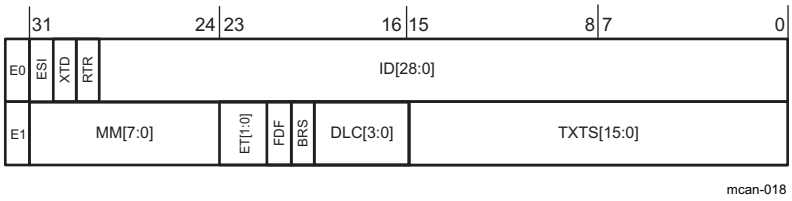


Table 22-115 shows Tx Event FIFO element field descriptions.

Table 22-115. Tx Event FIFO Element Field Descriptions

Word	Bits	Field Name	Description
E0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Data frame transmitted • 0x1: Remote frame transmitted
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

Table 22-115. Tx Event FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
E1	31:24	MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 22-114).
	23:22	ET[1:0]	Event Type • 0x0: Reserved • 0x1: Tx event • 0x2: Transmission in spite of cancellation (always set for transmissions in DAR mode) • 0x3: Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame transmitted without bit rate switching • 0x1: Frame transmitted with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: frame with 0-8 data bytes transmitted • 0x9-0xF (9-15): CAN: frame with 8 data bytes transmitted • 0x9-0xF (9-15): CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted
	15:0	TXTS[15:0]	Tx Timestamp Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC[19:16] TCP filed.

22.2.2.11.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address MCAN_SIDFC[15:2] FLSSA field plus the index of the filter element (0-127).

Figure 22-128 shows Standard Message ID Filter element structure.

Figure 22-128. Standard Message ID Filter Element Structure

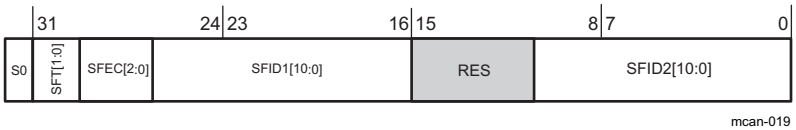


Table 22-116 shows Standard Message ID Filter element field descriptions.

Table 22-116. Standard Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
S0	31:30	SFT[1:0]	Standard Filter Type • 0x0: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) • 0x1: Dual ID filter for SFID1 or SFID2 • 0x2: Classic filter: SFID1 = filter; SFID2 = mask • 0x3: Filter element disabled **Note:** With SFT = '11' the filter element is disabled and the acceptance filtering continues (same behaviour as with SFEC = '000')
	29:27	SFEC[2:0]	Standard Filter Element Configuration All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = '100', '101', or '110' a match sets interrupt flag MCAN_IR[8]HPM and, if enabled, an interrupt is generated. In this case the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer, configuration of SFT[1:0] ignored
	26:16	SFID1[10:0]	Standard Filter ID 1 When filtering for Rx Buffers this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.
	15:11	RES	Reserved
	10:0	SFID2[10:0]	Standard Filter ID 2 This bit field has a different meaning depending on the configuration of SFEC: • 1) SFEC = '001' - '110' Second ID of standard ID filter element • 2) SFEC = '111' Filter for Rx Buffers
		SFID2[10:9]	This filed is decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		SFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		SFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC[15:2] RBSA field for storage of a matching message.

22.2.2.11.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC[15:2] FLESA field plus two times the index of the filter element (0-63).

Figure 22-129 shows Extended Message ID Filter element structure.

Figure 22-129. Extended Message ID Filter Element Structure

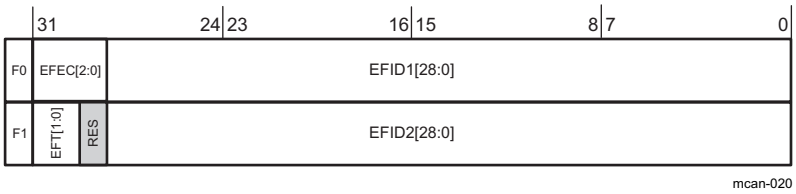


Table 22-117 shows Extended Message ID Filter element field descriptions.

Table 22-117. Extended Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
F0	31:29	EFEC[2:0]	Extended Filter Element Configuration All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = '100', '101', or '110' a match sets interrupt flag MCAN_IR[8]HPM and, if enabled, an interrupt is generated. In this case the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored
	28:0	EFID1[28:0]	Extended Filter ID 1 First ID of extended ID filter element. When filtering for Rx Buffers this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see Section 22.2.2.8.1.5, *Extended Message ID Filtering*) is used.

Table 22-117. Extended Message ID Filter Element Field Descriptions (continued)

Word	Bits	Field Name	Description
F1	31:30	EFT[1:0]	Extended Filter Type • 0x0: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) • 0x1: Dual ID filter for EFID1 or EFID2 • 0x2: Classic filter: EFID1 = filter, EFID2 = mask • 0x3: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), XIDAM mask not applied
	29	RES	Reserved
	28:0	EFID2[28:0]	Extended Filter ID 2 This bit field has a different meaning depending on the configuration of EFEC: • 1) EFEC = '001' - '110' Second ID of extended ID filter element • 2) EFEC = '111' Filter for Rx Buffers
		EFID2[10:9]	This field decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		EFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		EFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC[15:2] RBSA field for storage of a matching message.

22.2.3 MCAN Register Manual

22.2.3.1 MSS_MCAN_CFG Registers

Table 22-194 lists the MSS_MCAN_CFG registers. All register offset addresses not listed in Table 22-194 should be considered as reserved locations and the register contents should not be modified.

Table 22-118. MSS_MCAN_CFG Registers

Offset	Acronym	Register Name	Section
0h	SS_PID	SS_PID	Section 22.2.3.1.1
4h	SS_CTRL	SS_CTRL	Section 22.2.3.1.2
8h	SS_STAT	SS_STAT	Section 22.2.3.1.3
Ch	SS_ICS	SS_ICS	Section 22.2.3.1.4
10h	SS_IRS	SS_IRS	Section 22.2.3.1.5
14h	SS_IECS	SS_IECS	Section 22.2.3.1.6
18h	SS_IE	SS_IE	Section 22.2.3.1.7
1Ch	SS_IES	SS_IES	Section 22.2.3.1.8
20h	SS_EOI	SS_EOI	Section 22.2.3.1.9
24h	SS_EXT_TS_PS	SS_EXT_TS_PS	Section 22.2.3.1.10
28h	SS_EXT_TS_USIC	SS_EXT_TS_USIC	Section 22.2.3.1.11
200h	CREL	CREL	Section 22.2.3.1.12
204h	ENDN	ENDN	Section 22.2.3.1.13
208h	CUST	CUST	Section 22.2.3.1.14
20Ch	DBTP	DBTP	Section 22.2.3.1.15
210h	TEST	TEST	Section 22.2.3.1.16
214h	RWD	RWD	Section 22.2.3.1.17
218h	CCCR	CCCR	Section 22.2.3.1.18
21Ch	NBTP	NBTP	Section 22.2.3.1.19
220h	TSCC	TSCC	Section 22.2.3.1.20
224h	TSCV	TSCV	Section 22.2.3.1.21
228h	TOCC	TOCC	Section 22.2.3.1.22
22Ch	TOCV	TOCV	Section 22.2.3.1.23
230h	RES00	RES00	Section 22.2.3.1.24
234h	RES01	RES01	Section 22.2.3.1.25
238h	RES02	RES02	Section 22.2.3.1.26
23Ch	RES03	RES03	Section 22.2.3.1.27
240h	ECR	ECR	Section 22.2.3.1.28
244h	PSR	PSR	Section 22.2.3.1.29
248h	TDCR	TDCR	Section 22.2.3.1.30
24Ch	RES04	RES04	Section 22.2.3.1.31
250h	IR	IR	Section 22.2.3.1.32
254h	IE	IE	Section 22.2.3.1.33
258h	ILS	ILS	Section 22.2.3.1.34
25Ch	ILE	ILE	Section 22.2.3.1.35
260h	RES05	RES05	Section 22.2.3.1.36
264h	RES06	RES06	Section 22.2.3.1.37
268h	RES07	RES07	Section 22.2.3.1.38
26Ch	RES08	RES08	Section 22.2.3.1.39
270h	RES09	RES09	Section 22.2.3.1.40
274h	RES10	RES10	Section 22.2.3.1.41
278h	RES11	RES11	Section 22.2.3.1.42
27Ch	RES12	RES12	Section 22.2.3.1.43
280h	GFC	GFC	Section 22.2.3.1.44

Table 22-118. MSS_MCAN_CFG Registers (continued)

Offset	Acronym	Register Name	Section
284h	SIDFC	SIDFC	Section 22.2.3.1.45
288h	XIDFC	XIDFC	Section 22.2.3.1.46
28Ch	RES13	RES13	Section 22.2.3.1.47
290h	XIDAM	XIDAM	Section 22.2.3.1.48
294h	HPMS	HPMS	Section 22.2.3.1.49
298h	NDAT1	NDAT1	Section 22.2.3.1.50
29Ch	NDAT2	NDAT2	Section 22.2.3.1.51
2A0h	RXF0C	RXF0C	Section 22.2.3.1.52
2A4h	RXF0S	RXF0S	Section 22.2.3.1.53
2A8h	RXF0A	RXF0A	Section 22.2.3.1.54
2ACh	RXBC	RXBC	Section 22.2.3.1.55
2B0h	RXF1C	RXF1C	Section 22.2.3.1.56
2B4h	RXF1S	RXF1S	Section 22.2.3.1.57
2B8h	RXF1A	RXF1A	Section 22.2.3.1.58
2BCh	RXESC	RXESC	Section 22.2.3.1.59
2C0h	TXBC	TXBC	Section 22.2.3.1.60
2C4h	TXFQS	TXFQS	Section 22.2.3.1.61
2C8h	TXESC	TXESC	Section 22.2.3.1.62
2CCh	TXBRP	TXBRP	Section 22.2.3.1.63
2D0h	TXBAR	TXBAR	Section 22.2.3.1.64
2D4h	TXBCR	TXBCR	Section 22.2.3.1.65
2D8h	TXBTO	TXBTO	Section 22.2.3.1.66
2DCh	TXBCF	TXBCF	Section 22.2.3.1.67
2E0h	TXBTIE	TXBTIE	Section 22.2.3.1.68
2E4h	TXBCIE	TXBCIE	Section 22.2.3.1.69
2E8h	RES14	RES14	Section 22.2.3.1.70
2ECh	RES15	RES15	Section 22.2.3.1.71
2F0h	TXEFC	TXEFC	Section 22.2.3.1.72
2F4h	TXEFS	TXEFS	Section 22.2.3.1.73
2F8h	TXEFA	TXEFA	Section 22.2.3.1.74
2FCh	RES16	RES16	Section 22.2.3.1.75

22.2.3.1.1 SS_PID Register (Offset = 0h) [reset = 68E04101h]

SS_PID is shown in Figure 22-130 and described in Table 22-119.

Return to the Summary Table.

SS_PID

Figure 22-130. SS_PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		BU		MODULE_ID											
R-1h		R-2h		R-8E0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-8h					R-1h			R-0h		R-1h					

Table 22-119. SS_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID register scheme
29-28	BU	R	2h	Business Unit: 10 = Processors
27-16	MODULE_ID	R	8E0h	Module ID
15-11	RTL	R	8h	RTL revision. Will vary depending on release.
10-8	MAJOR	R	1h	Major revision
7-6	CUSTOM	R	0h	Custom
5-0	MINOR	R	1h	Minor revision

22.2.3.1.2 SS_CTRL Register (Offset = 4h) [reset = 8h]

SS_CTRL is shown in Figure 22-131 and described in Table 22-120.

Return to the Summary Table.

SS_CTRL

Figure 22-131. SS_CTRL Register

31	30	29	28	27	26	25	24
NU0							
R-0h							
23	22	21	20	19	18	17	16
NU0							
R-0h							
15	14	13	12	11	10	9	8
NU0							
R-0h							
7	6	5	4	3	2	1	0
NU0	EXT_TS_CNTR_EN	AUTOWAKEUP	WAKEUPREGEN	DBGSUSP_FREE	NU		
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R-0h		

Table 22-120. SS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	NU0	R	0h	Reserved
6	EXT_TS_CNTR_EN	R/W	0h	External TimeStamp Counter Enable
5	AUTOWAKEUP	R/W	0h	Automatic Wakeup Enable
4	WAKEUPREGEN	R/W	0h	Wakeup Request Enable
3	DBGSUSP_FREE	R/W	1h	0-Honor Debug Suspend, 1-Disregard debug suspend
2-0	NU	R	0h	Reserved

22.2.3.1.3 SS_STAT Register (Offset = 8h) [reset = 6h]

SS_STAT is shown in Figure 22-132 and described in Table 22-121.

Return to the Summary Table.

SS_STAT

Figure 22-132. SS_STAT Register

31	30	29	28	27	26	25	24
NU1							
R-0h							
23	22	21	20	19	18	17	16
NU1							
R-0h							
15	14	13	12	11	10	9	8
NU1							
R-0h							
7	6	5	4	3	2	1	0
NU1					EN_FDOE	MMI_DONE	NU
R-0h					R-1h	R-1h	R-0h

Table 22-121. SS_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU1	R	0h	Reserved
2	EN_FDOE	R	1h	Reflects the value of mcanss_enable_fdoe configuration port x=mcanss_enable_fdoe
1	MMI_DONE	R	1h	0:Memory Initialization is in progress, 1:Memory Intialization Done
0	NU	R	0h	Reserved

22.2.3.1.4 SS_ICS Register (Offset = Ch) [reset = 0h]

SS_ICS is shown in Figure 22-133 and described in Table 22-122.

Return to the Summary Table.

SS_ICS

Figure 22-133. SS_ICS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2															ICS
R-0h															W-0h

Table 22-122. SS_ICS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU2	R	0h	Reserved
0	ICS	W	0h	This bit contains the External TimeStamp Counter Overflow Interrupt status. Write '1' to clear bits. (ICS - Interrupt Clear Shadow Register)

22.2.3.1.5 SS_IRS Register (Offset = 10h) [reset = 0h]

SS_IRS is shown in Figure 22-134 and described in Table 22-123.

Return to the Summary Table.

SS_IRS

Figure 22-134. SS_IRS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3															IRS
R-0h															R-0h

Table 22-123. SS_IRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU3	R	0h	Reserved
0	IRS	R	0h	External TimeStamp Counter Overflow Interrupt status. Read raw interrupt status. (IRS - Interrupt Raw Status Register)

22.2.3.1.6 SS_IECS Register (Offset = 14h) [reset = 0h]

SS_IECS is shown in Figure 22-135 and described in Table 22-124.

Return to the Summary Table.

SS_IECS

Figure 22-135. SS_IECS Register

31	30	29	28	27	26	25	24
NU4							
R-0h							
23	22	21	20	19	18	17	16
NU4							
R-0h							
15	14	13	12	11	10	9	8
NU4							
R-0h							
7	6	5	4	3	2	1	0
NU4							IECS
R-0h							W-0h

Table 22-124. SS_IECS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU4	R	0h	Reserved
0	IECS	W	0h	External TimeStamp Counter Overflow Interrupt. Write '1' to clear bits. (IECS - Interrupt Enable Clear Shadow Register)

22.2.3.1.7 SS_IE Register (Offset = 18h) [reset = 0h]

SS_IE is shown in Figure 22-136 and described in Table 22-125.

Return to the Summary Table.

SS_IE

Figure 22-136. SS_IE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU5															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU5															IE
R-0h															R/W-0h

Table 22-125. SS_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU5	R	0h	Reserved
0	IE	R/W	0h	External TimeStamp Counter Overflow Interrupt. Write '1' to set interrupt enable. Read returns interrupt enable. (IE - Interrupt Enable Register)

22.2.3.1.8 SS_IES Register (Offset = 1Ch) [reset = 0h]

SS_IES is shown in Figure 22-137 and described in Table 22-126.

Return to the Summary Table.

SS_IES

Figure 22-137. SS_IES Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU6															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU6															IES
R-0h															R-0h

Table 22-126. SS_IES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU6	R	0h	Reserved
0	IES	R	0h	External TimeStamp Counter Overflow Interrupt. Read Enabled Interrupts. (IES - Interrupt Enable Status)

22.2.3.1.9 SS_EOI Register (Offset = 20h) [reset = 0h]

SS_EOI is shown in Figure 22-138 and described in Table 22-127.

Return to the Summary Table.

SS_EOI

Figure 22-138. SS_EOI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU7																								EOI							
R-0h																								W-0h							

Table 22-127. SS_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU7	R	0h	Reserved
7-0	EOI	W	0h	Write with bit position of targeted interrupt. (E.g. Ext TS is bit 0). Upon write, level interrupt will clear and if unserviced interrupt counter > 1 will issue another pulse interrupt. Field values: ext_ts_eoi(0): EOI value for External TS interrupt mcan_0_eoi(1): EOI value for mcan[0] interrupt mcan_1_eoi(2): EOI value for mcan[1] interrupt (EOI - End Of Interrupt)

22.2.3.1.10 SS_EXT_TS_PS Register (Offset = 24h) [reset = 0h]

SS_EXT_TS_PS is shown in Figure 22-139 and described in Table 22-128.

Return to the Summary Table.

SS_EXT_TS_PS

Figure 22-139. SS_EXT_TS_PS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU8								PRESCALE																							
R-0h								R/W-0h																							

Table 22-128. SS_EXT_TS_PS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU8	R	0h	Reserved
23-0	PRESCALE	R/W	0h	External Timestamp Prescaler reload value. External Timestamp count rate is host clock rate divided by this value with one exception: a value of 0 has the same effect as 1 .

22.2.3.1.11 SS_EXT_TS_USIC Register (Offset = 28h) [reset = 0h]

SS_EXT_TS_USIC is shown in Figure 22-140 and described in Table 22-129.

Return to the Summary Table.

SS_EXT_TS_USIC

Figure 22-140. SS_EXT_TS_USIC Register

31	30	29	28	27	26	25	24
NU9							
R-0h							
23	22	21	20	19	18	17	16
NU9							
R-0h							
15	14	13	12	11	10	9	8
NU9							
R-0h							
7	6	5	4	3	2	1	0
NU9			EXT_TS_INTR_CNTR				
R-0h			R-0h				

Table 22-129. SS_EXT_TS_USIC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	NU9	R	0h	Reserved
4-0	EXT_TS_INTR_CNTR	R	0h	Number of unserviced rollover interrupts. If >1 an EOI write will issue another pulse interrupt (EXT_TS_USIC - External TImeStamp Unserviced Interrupts Counter)

22.2.3.1.12 CREL Register (Offset = 200h) [reset = 32380608h]

CREL is shown in Figure 22-141 and described in Table 22-130.

Return to the Summary Table.

CREL

Figure 22-141. CREL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL				STEP				SUBSTEP				YEAR			
R-3h				R-2h				R-3h				R-8h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON								DAY							
R-6h								R-8h							

Table 22-130. CREL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	REL	R	3h	Core Release
27-24	STEP	R	2h	Step of Core Release
23-20	SUBSTEP	R	3h	Sub-Step of Core Release
19-16	YEAR	R	8h	Time Stamp Year
15-8	MON	R	6h	Time Stamp Month
7-0	DAY	R	8h	Time Stamp Day

22.2.3.1.13 ENDN Register (Offset = 204h) [reset = 87654321h]

ENDN is shown in Figure 22-142 and described in Table 22-131.

Return to the Summary Table.

ENDN

Figure 22-142. ENDN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV																															
R-87654321h																															

Table 22-131. ENDN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ETV	R	87654321h	Endianess test value

22.2.3.1.14 CUST Register (Offset = 208h) [reset = 0h]

CUST is shown in Figure 22-143 and described in Table 22-132.

Return to the Summary Table.

CUST

Figure 22-143. CUST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST																															
R-0h																															

Table 22-132. CUST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CUST	R	0h	Custom

22.2.3.1.15 DBTP Register (Offset = 20Ch) [reset = A33h]

DBTP is shown in Figure 22-144 and described in Table 22-133.

Return to the Summary Table.

DBTP

Figure 22-144. DBTP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU13								TDC	NU12		DBRP				
R-0h								R/W-0h	R-0h		R/W-0h				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU11			DTSEG1					DTSEG2				DSJW			
R-0h			R/W-Ah					R/W-3h				R/W-3h			

Table 22-133. DBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU13	R	0h	Reserved
23	TDC	R/W	0h	Transmitter Delay Compensation
22-21	NU12	R	0h	Reserved
20-16	DBRP	R/W	0h	Data Baud Rate Prescaler
15-13	NU11	R	0h	Reserved
12-8	DTSEG1	R/W	Ah	Data time segment before smaple point
7-4	DTSEG2	R/W	3h	Data time segment after sample point
3-0	DSJW	R/W	3h	Data resynchronization Jump Width

22.2.3.1.16 TEST Register (Offset = 210h) [reset = 0h]

TEST is shown in Figure 22-145 and described in Table 22-134.

Return to the Summary Table.

TEST

Figure 22-145. TEST Register

31	30	29	28	27	26	25	24
NU15							
R-0h							
23	22	21	20	19	18	17	16
NU15							
R-0h							
15	14	13	12	11	10	9	8
NU15							
R-0h							
7	6	5	4	3	2	1	0
RX	TX		LBCK	NU14			
R-0h	R/W-0h		R/W-0h	R-0h			

Table 22-134. TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU15	R	0h	Reserved
7	RX	R	0h	Receive Pin
6-5	TX	R/W	0h	Control of Transmit Pin
4	LBCK	R/W	0h	Loop Back Mode
3-0	NU14	R	0h	Reserved

22.2.3.1.17 RWD Register (Offset = 214h) [reset = 0h]

RWD is shown in Figure 22-146 and described in Table 22-135.

Return to the Summary Table.

RWD

Figure 22-146. RWD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU16																WDV								WDC							
R-0h																R-0h								R/W-0h							

Table 22-135. RWD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU16	R	0h	Reserved
15-8	WDV	R	0h	Watchdog Value
7-0	WDC	R/W	0h	Watchdog Counter Value

22.2.3.1.18 CCCR Register (Offset = 218h) [reset = 1h]

CCCR is shown in Figure 22-147 and described in Table 22-136.

Return to the Summary Table.

CCCR

Figure 22-147. CCCR Register

31	30	29	28	27	26	25	24
NU18							
R/W-0h							
23	22	21	20	19	18	17	16
NU18							
R/W-0h							
15	14	13	12	11	10	9	8
NU18	TXP	EFBI	PXHD	NU17		BRSE	FDOE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-1h

Table 22-136. CCCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	NU18	R/W	0h	Reserved
14	TXP	R/W	0h	Transmit Pause
13	EFBI	R/W	0h	Edge Filtering durign Bus Integration
12	PXHD	R/W	0h	Protocol Exception Handling Disable
11-10	NU17	R	0h	Reserved
9	BRSE	R/W	0h	Bit Rate Switch Enable
8	FDOE	R/W	0h	FD Operation Enable
7	TEST	R/W	0h	Test Mode enable
6	DAR	R/W	0h	Disable Automatic Regransmission
5	MON	R/W	0h	Bus Monitoring Mode
4	CSR	R/W	0h	Clock Stop Request
3	CSA	R	0h	Clock Stop Acknowledge
2	ASM	R/W	0h	Restriced Operation Mode
1	CCE	R/W	0h	Configuration Change Enable
0	INIT	R/W	1h	Initialization

22.2.3.1.19 NBTP Register (Offset = 21Ch) [reset = 06000A03h]

NBTP is shown in Figure 22-148 and described in Table 22-137.

Return to the Summary Table.

NBTP

Figure 22-148. NBTP Register

31	30	29	28	27	26	25	24
NSJW							NBRP
R/W-3h							R/W-0h
23	22	21	20	19	18	17	16
NBRP							
R/W-0h							
15	14	13	12	11	10	9	8
NTSEG1							
R/W-Ah							
7	6	5	4	3	2	1	0
NU19	NTSEG2						
R-0h	R/W-3h						

Table 22-137. NBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NSJW	R/W	3h	Nominal Resynchronization Jump Width
24-16	NBRP	R/W	0h	Nominal Baud Rate Prescaler
15-8	NTSEG1	R/W	Ah	Nominal Time segment before sample point
7	NU19	R	0h	Reserved
6-0	NTSEG2	R/W	3h	Nominal Time segment after sample point

22.2.3.1.20 TSCC Register (Offset = 220h) [reset = 0h]

TSCC is shown in Figure 22-149 and described in Table 22-138.

Return to the Summary Table.

TSCC

Figure 22-149. TSCC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU21												TCP			
R-0h												R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU20														TSS	
R-0h														R/W-0h	

Table 22-138. TSCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU21	R	0h	Reserved
19-16	TCP	R/W	0h	Timestamp Counter Prescaler
15-2	NU20	R	0h	Reserved
1-0	TSS	R/W	0h	Timestamp Select

22.2.3.1.21 TSCV Register (Offset = 224h) [reset = 0h]

TSCV is shown in Figure 22-150 and described in Table 22-139.

Return to the Summary Table.

TSCV

Figure 22-150. TSCV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU22																TSC															
R-0h																R/W-0h															

Table 22-139. TSCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU22	R	0h	Reserved
15-0	TSC	R/W	0h	Timestamp Counter

22.2.3.1.22 TOCC Register (Offset = 228h) [reset = FFFF0000h]

TOCC is shown in Figure 22-151 and described in Table 22-140.

Return to the Summary Table.

TOCC

Figure 22-151. TOCC Register

31	30	29	28	27	26	25	24
TOP							
R/W-FFFFh							
23	22	21	20	19	18	17	16
TOP							
R/W-FFFFh							
15	14	13	12	11	10	9	8
NU23							
R-0h							
7	6	5	4	3	2	1	0
NU23					TOS		ETOC
R-0h					R/W-0h		R/W-0h

Table 22-140. TOCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TOP	R/W	FFFFh	Timeout Period
15-3	NU23	R	0h	Reserved
2-1	TOS	R/W	0h	Timeout Select
0	ETOC	R/W	0h	Enable Timeout Counter

22.2.3.1.23 TOCV Register (Offset = 22Ch) [reset = FFFFh]

TOCV is shown in Figure 22-152 and described in Table 22-141.

Return to the Summary Table.

TOCV

Figure 22-152. TOCV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU24																TOC															
R-0h																R/W-FFFFh															

Table 22-141. TOCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU24	R	0h	Reserved
15-0	TOC	R/W	FFFFh	Timeout Counter

22.2.3.1.24 RES00 Register (Offset = 230h) [reset = 0h]

RES00 is shown in Figure 22-153 and described in Table 22-142.

Return to the Summary Table.

RES00

Figure 22-153. RES00 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES00																															
R-0h																															

Table 22-142. RES00 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES00	R	0h	Reserved

22.2.3.1.25 RES01 Register (Offset = 234h) [reset = 0h]

RES01 is shown in Figure 22-154 and described in Table 22-143.

Return to the Summary Table.

RES01

Figure 22-154. RES01 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES01																															
R-0h																															

Table 22-143. RES01 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES01	R	0h	Reserved

22.2.3.1.26 RES02 Register (Offset = 238h) [reset = 0h]

RES02 is shown in Figure 22-155 and described in Table 22-144.

Return to the Summary Table.

RES02

Figure 22-155. RES02 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES02																															
R-0h																															

Table 22-144. RES02 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES02	R	0h	Reserved

22.2.3.1.27 RES03 Register (Offset = 23Ch) [reset = 0h]

RES03 is shown in Figure 22-156 and described in Table 22-145.

Return to the Summary Table.

RES03

Figure 22-156. RES03 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES03																															
R-0h																															

Table 22-145. RES03 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES03	R	0h	Reserved

22.2.3.1.28 ECR Register (Offset = 240h) [reset = 0h]

ECR is shown in Figure 22-157 and described in Table 22-146.

Return to the Summary Table.

ECR

Figure 22-157. ECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU25								CEL							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC							TEC							
R-0h	R-0h							R-0h							

Table 22-146. ECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU25	R	0h	Reserved
23-16	CEL	R	0h	CAN Error Logging
15	RP	R	0h	Recieve Error Passive
14-8	REC	R	0h	Recieve Error Counter
7-0	TEC	R	0h	Transmit Error Counter

22.2.3.1.29 PSR Register (Offset = 244h) [reset = 707h]

PSR is shown in Figure 22-158 and described in Table 22-147.

Return to the Summary Table.

PSR

Figure 22-158. PSR Register

31	30	29	28	27	26	25	24
NU27							
R-0h							
23	22	21	20	19	18	17	16
NU27	TDCV						
R-0h	R-0h						
15	14	13	12	11	10	9	8
NU26	PXE	RFDF	RBRS	RESI	DLEC		
R-0h	R-0h	R-0h	R-0h	R-0h	R-7h		
7	6	5	4	3	2	1	0
BO	EW	EP	ACT		LEC		
R-0h	R-0h	R-0h	R-0h		R-7h		

Table 22-147. PSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	NU27	R	0h	Reserved
22-16	TDCV	R	0h	Transmitter Delay Compensation Value
15	NU26	R	0h	Reserved
14	PXE	R	0h	Protocol Exception Event
13	RFDF	R	0h	Recieved a CAN FD Message
12	RBRS	R	0h	BRS flag of last recieved CAN FD Message
11	RESI	R	0h	ESI flag of last recieved CAN FD Message
10-8	DLEC	R	7h	Data Phase Last Error Code
7	BO	R	0h	Bus_Off status
6	EW	R	0h	Warning Status
5	EP	R	0h	Error Passive
4-3	ACT	R	0h	Activity
2-0	LEC	R	7h	Last Error Code

22.2.3.1.30 TDCR Register (Offset = 248h) [reset = 0h]

TDCR is shown in Figure 22-159 and described in Table 22-148.

Return to the Summary Table.

TDCR

Figure 22-159. TDCR Register

31	30	29	28	27	26	25	24
NU29							
R-0h							
23	22	21	20	19	18	17	16
NU29							
R-0h							
15	14	13	12	11	10	9	8
NU29	TDCO						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
NU28	TDCF						
R-0h	R/W-0h						

Table 22-148. TDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	NU29	R	0h	Reserved
14-8	TDCO	R/W	0h	Transmitter Delay Compensation Offset
7	NU28	R	0h	Reserved
6-0	TDCF	R/W	0h	Transmitter Delay Compensation Filter Window Length

22.2.3.1.31 RES04 Register (Offset = 24Ch) [reset = 0h]

RES04 is shown in Figure 22-160 and described in Table 22-149.

Return to the Summary Table.

RES04

Figure 22-160. RES04 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES04																															
R-0h																															

Table 22-149. RES04 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES04	R	0h	Reserved

22.2.3.1.32 IR Register (Offset = 250h) [reset = 0h]

IR is shown in Figure 22-161 and described in Table 22-150.

Return to the Summary Table.

IR

Figure 22-161. IR Register

31	30	29	28	27	26	25	24
NU30		ARA	PED	PEA	WDI	BO	EW
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-150. IR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU30	R	0h	Reserved
29	ARA	R/W	0h	Access to Reserved Address
28	PED	R/W	0h	Protocol Error in data Phase
27	PEA	R/W	0h	Protocol Error in Arbitration Phase
26	WDI	R/W	0h	Watchdog Interrupt
25	BO	R/W	0h	Bus_Off Status
24	EW	R/W	0h	Warning Status
23	EP	R/W	0h	Error Passive
22	ELO	R/W	0h	Error Logging Overflow
21	BEU	R/W	0h	Bit Error Uncorrected
20	BEC	R/W	0h	Bit Error Corrected
19	DRX	R/W	0h	Message stored to Dedicated Rx Buffer
18	TOO	R/W	0h	Timeout Occurred
17	MRAF	R/W	0h	Message RAM Access Failure
16	TSW	R/W	0h	Timestamp Wraparound
15	TEFL	R/W	0h	Tx Event FIFO Element Lost
14	TEFF	R/W	0h	Tx Event FIFO Full
13	TEFW	R/W	0h	Tx Event FIFO Watermark Reached
12	TEFN	R/W	0h	Tx Event FIFO New Entry
11	TFE	R/W	0h	Tx FIFO Empty
10	TCF	R/W	0h	Transmission Cancellation Finished
9	TC	R/W	0h	Transmission Complete
8	HPM	R/W	0h	High Priority Message
7	RF1L	R/W	0h	Rx FIFO 1 Message Lost

Table 22-150. IR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RF1F	R/W	0h	Rx FIFO 1 Full
5	RF1W	R/W	0h	Rx FIFO 1 Watermark Reached
4	RF1N	R/W	0h	Rx FIFO 1 New Message
3	RF0L	R/W	0h	Rx FIFO 0 Message Lost
2	RF0F	R/W	0h	Rx FIFO 0 Full
1	RF0W	R/W	0h	Rx FIFO 0 Watermark Reached
0	RF0N	R/W	0h	Rx FIFO 0 New Message

22.2.3.1.33 IE Register (Offset = 254h) [reset = 0h]

IE is shown in Figure 22-162 and described in Table 22-151.

Return to the Summary Table.

IE

Figure 22-162. IE Register

31	30	29	28	27	26	25	24
NU31		ARAE	PEDE	PEAE	WDIE	BOE	EWE
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPE	ELOE	BEUE	BECE	DRX	TOOE	MRAFE	TSWE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-151. IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU31	R	0h	Reserved
29	ARAE	R/W	0h	Accees to Reserve Address Interrupt Enable
28	PEDE	R/W	0h	Protocol Error in Data Phase Interrupt Enable
27	PEAE	R/W	0h	Protocol Error in Arbitration Phase Interrupt Enable
26	WDIE	R/W	0h	Watchdog Interrupt Enable
25	BOE	R/W	0h	Bus_Off Status Interrupt Enable
24	EWE	R/W	0h	Warning Status Interrupt Enable
23	EPE	R/W	0h	Error Passive Interrupt Enable
22	ELOE	R/W	0h	Error Logging Overflow Interrupt Enable
21	BEUE	R/W	0h	Bit Error Uncorrected Interrupt Enable
20	BECE	R/W	0h	Bit Error Corrected Interrupt Enable
19	DRX	R/W	0h	Message stored to Dedicated Rx Buffer Interrupt Enable
18	TOOE	R/W	0h	Timeout Occurred Interrupt Enable
17	MRAFE	R/W	0h	Message RAM Access Failure Interrupt Enable
16	TSWE	R/W	0h	Timestamp Wraparound Interrupt Enable
15	TEFLE	R/W	0h	Tx Event FIFO Event Lost Interrupt Enable
14	TEFFE	R/W	0h	Tx Event FIFO Full Interrupt Enable
13	TEFWE	R/W	0h	Tx Event FIFO Watermark Reached Interrupt enable
12	TEFNE	R/W	0h	Tx Event FIFO New Entry Interrupt Enable
11	TFEE	R/W	0h	Tx FIFO Empty Interrupt Enable
10	TCFE	R/W	0h	Transmission Cancellation Finishied Interrupt Enable
9	TCE	R/W	0h	Transmission Completed Interrupt Enable
8	HPME	R/W	0h	High Priority message Interrupt Enable
7	RF1LE	R/W	0h	rx FIFO 1 Message Lost Interrupt Enable

Table 22-151. IE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RF1FE	R/W	0h	Rx FIFO 1 Full Interrupt Enable
5	RF1WE	R/W	0h	Rx FIFO 1 Watermark Reached Interrupt Enable
4	RF1NE	R/W	0h	Rx FIFO 1 New Message Interrupt Enable
3	RF0LE	R/W	0h	Rx FIFO 0 Message Lost Interrupt Enable
2	RF0FE	R/W	0h	Rx FIFO 0 Full Interrupt Enable
1	RF0WE	R/W	0h	Rx FIFO 0 Watermark Reached Interrupt Enable
0	RF0NE	R/W	0h	Rx FIFO 0 New Message Interrupt Enable

22.2.3.1.34 ILS Register (Offset = 258h) [reset = 0h]

ILS is shown in Figure 22-163 and described in Table 22-152.

Return to the Summary Table.

ILS

Figure 22-163. ILS Register

31	30	29	28	27	26	25	24
NU32		ARAL	PEDL	PEAL	WDIL	BOL	EWL
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-152. ILS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU32	R	0h	Reserved
29	ARAL	R/W	0h	Accees to Reserve Address Interrupt Line
28	PEDL	R/W	0h	Protocol Error in Data Phase Interrupt Line
27	PEAL	R/W	0h	Protocol Error in Arbitration Phase Interrupt Line
26	WDIL	R/W	0h	Watchdog Interrupt Line
25	BOL	R/W	0h	Bus_Off Status Interrupt Line
24	EWL	R/W	0h	Warning Status Interrupt Line
23	EPL	R/W	0h	Error Passive Interrupt Line
22	ELOL	R/W	0h	Error Logging Overflow Interrupt Line
21	BEUL	R/W	0h	Bit Error Uncorrected Interrupt Line
20	BECL	R/W	0h	Bit Error Corrected Interrupt Line
19	DRXL	R/W	0h	Message stored to Dedicated Rx Buffer Interrupt Line
18	TOOL	R/W	0h	Timeout Occurred Interrupt Line
17	MRAFL	R/W	0h	Message RAM Access Failure Interrupt Line
16	TSWL	R/W	0h	Timestamp Wraparound Interrupt Line
15	TEFLL	R/W	0h	Tx Event FIFO Event Lost Interrupt Line
14	TEFFL	R/W	0h	Tx Event FIFO Full Interrupt Line
13	TEFWL	R/W	0h	Tx Event FIFO Watermark Reached Interrupt Line
12	TEFNL	R/W	0h	Tx Event FIFO New Entry Interrupt Line
11	TFEL	R/W	0h	Tx FIFO Empty Interrupt Line
10	TCFL	R/W	0h	Transmission Cancellation Finishied Interrupt Line
9	TCL	R/W	0h	Transmission Completed Interrupt Line
8	HPML	R/W	0h	High Priority message Interrupt Line
7	RF1LL	R/W	0h	rx FIFO 1 Message Lost Interrupt Line

Table 22-152. ILS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RF1FL	R/W	0h	Rx FIFO 1 Full Interrupt Line
5	RF1WL	R/W	0h	Rx FIFO 1 Watermark Reached Interrupt Line
4	RF1NL	R/W	0h	Rx FIFO 1 New Message Interrupt Line
3	RF0LL	R/W	0h	Rx FIFO 0 Message Lost Interrupt Line
2	RF0FL	R/W	0h	Rx FIFO 0 Full Interrupt Line
1	RF0WL	R/W	0h	Rx FIFO 0 Watermark Reached Interrupt Line
0	RF0NL	R/W	0h	Rx FIFO 0 New Message Interrupt Line

22.2.3.1.35 ILE Register (Offset = 25Ch) [reset = 0h]

ILE is shown in Figure 22-164 and described in Table 22-153.

Return to the Summary Table.

ILE

Figure 22-164. ILE Register

31	30	29	28	27	26	25	24
NU33							
R-0h							
23	22	21	20	19	18	17	16
NU33							
R-0h							
15	14	13	12	11	10	9	8
NU33							
R-0h							
7	6	5	4	3	2	1	0
NU33						EINT1	EINT0
R-0h						R/W-0h	R/W-0h

Table 22-153. ILE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU33	R	0h	Reserved
1	EINT1	R/W	0h	Enable Interrupt Line 1
0	EINT0	R/W	0h	Enable Interrupt Line 0

22.2.3.1.36 RES05 Register (Offset = 260h) [reset = 0h]

RES05 is shown in Figure 22-165 and described in Table 22-154.

Return to the Summary Table.

RES05

Figure 22-165. RES05 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES05																															
R-0h																															

Table 22-154. RES05 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES05	R	0h	Reserved

22.2.3.1.37 RES06 Register (Offset = 264h) [reset = 0h]

RES06 is shown in Figure 22-166 and described in Table 22-155.

Return to the Summary Table.

RES06

Figure 22-166. RES06 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES06																															
R-0h																															

Table 22-155. RES06 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES06	R	0h	Reserved

22.2.3.1.38 RES07 Register (Offset = 268h) [reset = 0h]

RES07 is shown in Figure 22-167 and described in Table 22-156.

Return to the Summary Table.

RES07

Figure 22-167. RES07 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES07																															
R-0h																															

Table 22-156. RES07 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES07	R	0h	Reserved

22.2.3.1.39 RES08 Register (Offset = 26Ch) [reset = 0h]

RES08 is shown in Figure 22-168 and described in Table 22-157.

Return to the Summary Table.

RES08

Figure 22-168. RES08 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES08																															
R-0h																															

Table 22-157. RES08 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES08	R	0h	Reserved

22.2.3.1.40 RES09 Register (Offset = 270h) [reset = 0h]

RES09 is shown in Figure 22-169 and described in Table 22-158.

Return to the Summary Table.

RES09

Figure 22-169. RES09 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES09																															
R-0h																															

Table 22-158. RES09 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES09	R	0h	Reserved

22.2.3.1.41 RES10 Register (Offset = 274h) [reset = 0h]

RES10 is shown in Figure 22-170 and described in Table 22-159.

Return to the Summary Table.

RES10

Figure 22-170. RES10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES10																															
R-0h																															

Table 22-159. RES10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES10	R	0h	Reserved

22.2.3.1.42 RES11 Register (Offset = 278h) [reset = 0h]

RES11 is shown in Figure 22-171 and described in Table 22-160.

Return to the Summary Table.

RES11

Figure 22-171. RES11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES11																															
R-0h																															

Table 22-160. RES11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES11	R	0h	Reserved

22.2.3.1.43 RES12 Register (Offset = 27Ch) [reset = 0h]

RES12 is shown in Figure 22-172 and described in Table 22-161.

Return to the Summary Table.

RES12

Figure 22-172. RES12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES12																															
R-0h																															

Table 22-161. RES12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES12	R	0h	Reserved

22.2.3.1.44 GFC Register (Offset = 280h) [reset = 0h]

GFC is shown in Figure 22-173 and described in Table 22-162.

Return to the Summary Table.

GFC

Figure 22-173. GFC Register

31	30	29	28	27	26	25	24
NU34							
R-0h							
23	22	21	20	19	18	17	16
NU34							
R-0h							
15	14	13	12	11	10	9	8
NU34							
R-0h							
7	6	5	4	3	2	1	0
NU34		ANFS		ANFE		RRFS	RRFE
R-0h		R/W-0h		R/W-0h		R/W-0h	R/W-0h

Table 22-162. GFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU34	R	0h	Reserved
5-4	ANFS	R/W	0h	Accept Non-matching Frames Standard
3-2	ANFE	R/W	0h	Accept Non-matching Frames Extended
1	RRFS	R/W	0h	reject Remote Frames Standard
0	RRFE	R/W	0h	reject Remote Frames Extended

22.2.3.1.45 SIDFC Register (Offset = 284h) [reset = 0h]

SIDFC is shown in Figure 22-174 and described in Table 22-163.

Return to the Summary Table.

SIDFC

Figure 22-174. SIDFC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU36								LSS_S							
R-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLSSA_S														NU35	
R/W-0h														R-0h	

Table 22-163. SIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU36	R	0h	Reserved
23-16	LSS_S	R/W	0h	List Size Standard
15-2	FLSSA_S	R/W	0h	Filter List Standard Start Address
1-0	NU35	R	0h	Reserved

22.2.3.1.46 XIDFC Register (Offset = 288h) [reset = 0h]

XIDFC is shown in Figure 22-175 and described in Table 22-164.

Return to the Summary Table.

XIDFC

Figure 22-175. XIDFC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU38								LSS_X							
R-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLSSA_X														NU37	
R/W-0h														R-0h	

Table 22-164. XIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU38	R	0h	Reserved
23-16	LSS_X	R/W	0h	List Size Standard
15-2	FLSSA_X	R/W	0h	Filter List Standard Start Address
1-0	NU37	R	0h	Reserved

22.2.3.1.47 RES13 Register (Offset = 28Ch) [reset = 0h]

RES13 is shown in Figure 22-176 and described in Table 22-165.

Return to the Summary Table.

RES13

Figure 22-176. RES13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES13																															
R-0h																															

Table 22-165. RES13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES13	R	0h	Reserved

22.2.3.1.48 XIDAM Register (Offset = 290h) [reset = 1FFFFFFFh]

XIDAM is shown in Figure 22-177 and described in Table 22-166.

Return to the Summary Table.

XIDAM

Figure 22-177. XIDAM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU39			EIDM												
R-0h			R/W-1FFFFFFFh												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIDM															
R/W-1FFFFFFFh															

Table 22-166. XIDAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU39	R	0h	Reserved
28-0	EIDM	R/W	1FFFFFFFh	Extended ID Mask

22.2.3.1.49 HPMS Register (Offset = 294h) [reset = 0h]

HPMS is shown in Figure 22-178 and described in Table 22-167.

Return to the Summary Table.

HPMS

Figure 22-178. HPMS Register

31	30	29	28	27	26	25	24
NU40							
R-0h							
23	22	21	20	19	18	17	16
NU40							
R-0h							
15	14	13	12	11	10	9	8
FLST	FIDX						
R-0h	R-0h						
7	6	5	4	3	2	1	0
MSI		BIDX					
R-0h		R-0h					

Table 22-167. HPMS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU40	R	0h	Reserved
15	FLST	R	0h	Filter List
14-8	FIDX	R	0h	Filter Index
7-6	MSI	R	0h	Message Storeage Indicator
5-0	BIDX	R	0h	Buffer Index

22.2.3.1.50 NDAT1 Register (Offset = 298h) [reset = 0h]

NDAT1 is shown in Figure 22-179 and described in Table 22-168.

Return to the Summary Table.

NDAT1

Figure 22-179. NDAT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND0_31																															
R/W-0h																															

Table 22-168. NDAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ND0_31	R/W	0h	New Data 0-31

22.2.3.1.51 NDAT2 Register (Offset = 29Ch) [reset = 0h]

NDAT2 is shown in Figure 22-180 and described in Table 22-169.

Return to the Summary Table.

NDAT2

Figure 22-180. NDAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND32_63																															
R/W-0h																															

Table 22-169. NDAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ND32_63	R/W	0h	New Data 32-63

22.2.3.1.52 RXF0C Register (Offset = 2A0h) [reset = 0h]

RXF0C is shown in Figure 22-181 and described in Table 22-170.

Return to the Summary Table.

RXF0C

Figure 22-181. RXF0C Register

31	30	29	28	27	26	25	24
F0OM	F0WM						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
NU42_1	F0S						
R-0h	R/W-0h						
15	14	13	12	11	10	9	8
NU42	F0SA						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
F0SA						NU41	
R/W-0h						R-0h	

Table 22-170. RXF0C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F0OM	R/W	0h	Rx FIFO 0 Operation Mode
30-24	F0WM	R/W	0h	Rx FIFO 0 Watermark
23	NU42_1	R	0h	Reserved
22-16	F0S	R/W	0h	Rx FIFO 0 Size
15	NU42	R	0h	Reserved
14-2	F0SA	R/W	0h	Rx FIFO 0 Start Address
1-0	NU41	R	0h	Reserved

22.2.3.1.53 RXF0S Register (Offset = 2A4h) [reset = 0h]

RXF0S is shown in Figure 22-182 and described in Table 22-171.

Return to the Summary Table.

RXF0S

Figure 22-182. RXF0S Register

31	30	29	28	27	26	25	24
NU46						RF0L	F0F
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
NU45		F0PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
NU44		F0GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
NU43	F0FL						
R-0h	R-0h						

Table 22-171. RXF0S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU46	R	0h	Reserved
25	RF0L	R	0h	Rx FIFO 0 Message Lost
24	F0F	R	0h	Rx FIFO 0 Full
23-22	NU45	R	0h	Reserved
21-16	F0PI	R	0h	Rx FIFO 0 Put Index
15-14	NU44	R	0h	Reserved
13-8	F0GI	R	0h	Rx FIFO 0 Get Index
7	NU43	R	0h	Reserved
6-0	F0FL	R	0h	Rx FIFO 0 Fill Level

22.2.3.1.54 RXF0A Register (Offset = 2A8h) [reset = 0h]

RXF0A is shown in Figure 22-183 and described in Table 22-172.

Return to the Summary Table.

RXF0A

Figure 22-183. RXF0A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU47																										F0AI					
R-0h																										R/W-0h					

Table 22-172. RXF0A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU47	R	0h	Reserved
5-0	F0AI	R/W	0h	Rx FIFO 0 Acknowledge Index

22.2.3.1.55 RXBC Register (Offset = 2ACh) [reset = 0h]

RXBC is shown in Figure 22-184 and described in Table 22-173.

Return to the Summary Table.

RXBC

Figure 22-184. RXBC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU49															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBSA														NU48	
R/W-0h														R-0h	

Table 22-173. RXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU49	R	0h	Reserved
15-2	RBSA	R/W	0h	Rx Buffer Start Address
1-0	NU48	R	0h	Reserved

22.2.3.1.56 RXF1C Register (Offset = 2B0h) [reset = 0h]

RXF1C is shown in Figure 22-185 and described in Table 22-174.

Return to the Summary Table.

RXF1C

Figure 22-185. RXF1C Register

31	30	29	28	27	26	25	24
F1OM	F1WM						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
NU50_1	F1S						
R-0h	R/W-0h						
15	14	13	12	11	10	9	8
NU50	F1SA						
R-0h	R/W-0h						
7	6	5	4	3	2	1	0
F1SA						NU499	
R/W-0h						R-0h	

Table 22-174. RXF1C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F1OM	R/W	0h	Rx FIFO 0 Operation Mode
30-24	F1WM	R/W	0h	Rx FIFO 0 Watermark
23	NU50_1	R	0h	Reserved
22-16	F1S	R/W	0h	Rx FIFO 0 Size
15	NU50	R	0h	Reserved
14-2	F1SA	R/W	0h	Rx FIFO 0 Start Address
1-0	NU499	R	0h	Reserved

22.2.3.1.57 RXF1S Register (Offset = 2B4h) [reset = 0h]

RXF1S is shown in Figure 22-186 and described in Table 22-175.

Return to the Summary Table.

RXF1S

Figure 22-186. RXF1S Register

31	30	29	28	27	26	25	24
NU54						RF1L	F1F
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
NU53		F1PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
NU52		F1GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
NU51	F1FL						
R-0h	R-0h						

Table 22-175. RXF1S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU54	R	0h	Reserved
25	RF1L	R	0h	Rx FIFO 0 Message Lost
24	F1F	R	0h	Rx FIFO 0 Full
23-22	NU53	R	0h	Reserved
21-16	F1PI	R	0h	Rx FIFO 0 Put Index
15-14	NU52	R	0h	Reserved
13-8	F1GI	R	0h	Rx FIFO 0 Get Index
7	NU51	R	0h	Reserved
6-0	F1FL	R	0h	Rx FIFO 0 Fill Level

22.2.3.1.58 RXF1A Register (Offset = 2B8h) [reset = 0h]

RXF1A is shown in Figure 22-187 and described in Table 22-176.

Return to the Summary Table.

RXF1A

Figure 22-187. RXF1A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU55																										F1AI					
R-0h																										R/W-0h					

Table 22-176. RXF1A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU55	R	0h	Reserved
5-0	F1AI	R/W	0h	Rx FIFO 0 Acknowledge Index

22.2.3.1.59 RXESC Register (Offset = 2BCh) [reset = 0h]

RXESC is shown in Figure 22-188 and described in Table 22-177.

Return to the Summary Table.

RXESC

Figure 22-188. RXESC Register

31	30	29	28	27	26	25	24
NU58							
R-0h							
23	22	21	20	19	18	17	16
NU58							
R-0h							
15	14	13	12	11	10	9	8
NU58					RBDS		
R-0h					R/W-0h		
7	6	5	4	3	2	1	0
NU57	F1DS			NU56	F0DS		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 22-177. RXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU58	R	0h	Reserved
10-8	RBDS	R/W	0h	Rx Buffer data Field Size
7	NU57	R	0h	Reserved
6-4	F1DS	R/W	0h	Rx FIFO 1 Data Field Size
3	NU56	R	0h	Reserved
2-0	F0DS	R/W	0h	Rx FIFO 0 Data Field Size

22.2.3.1.60 TXBC Register (Offset = 2C0h) [reset = 0h]

TXBC is shown in Figure 22-189 and described in Table 22-178.

Return to the Summary Table.

TXBC

Figure 22-189. TXBC Register

31	30	29	28	27	26	25	24
NU61	TFQM	TFQS					
R-0h	R-0h	R-0h					
23	22	21	20	19	18	17	16
NU60		NDTB					
R-0h		R-0h					
15	14	13	12	11	10	9	8
TBSA							
R-0h							
7	6	5	4	3	2	1	0
TBSA						NU59	
R-0h						R-0h	

Table 22-178. TXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU61	R	0h	Reserved
30	TFQM	R	0h	Tx FIFO/Queue Mode
29-24	TFQS	R	0h	Transmit FIFO/Queue Size
23-22	NU60	R	0h	Reserved
21-16	NDTB	R	0h	Number of Dedicated Transmit Buffers
15-2	TBSA	R	0h	Tx Buffers Start Address
1-0	NU59	R	0h	Reserved

22.2.3.1.61 TXFQS Register (Offset = 2C4h) [reset = 0h]

TXFQS is shown in Figure 22-190 and described in Table 22-179.

Return to the Summary Table.

TXFQS

Figure 22-190. TXFQS Register

31	30	29	28	27	26	25	24
NU64							
R-0h							
23	22	21	20	19	18	17	16
NU64		TFQF	TFQPI				
R-0h		R-0h	R-0h				
15	14	13	12	11	10	9	8
NU63			TFGI				
R-0h			R-0h				
7	6	5	4	3	2	1	0
NU62		TFFL					
R-0h		R-0h					

Table 22-179. TXFQS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	NU64	R	0h	Reserved
21	TFQF	R	0h	Tx FIFO/Queue Full
20-16	TFQPI	R	0h	Tx FIFO/Queue Put Index
15-13	NU63	R	0h	Reserved
12-8	TFGI	R	0h	Tx Queue Get Index
7-6	NU62	R	0h	Reserved
5-0	TFFL	R	0h	Tx FIFO Free Level

22.2.3.1.62 TXESC Register (Offset = 2C8h) [reset = 0h]

TXESC is shown in Figure 22-191 and described in Table 22-180.

Return to the Summary Table.

TXESC

Figure 22-191. TXESC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU65															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU65													TBDS		
R-0h													R/W-0h		

Table 22-180. TXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU65	R	0h	Reserved
2-0	TBDS	R/W	0h	Tx Buffer Data Field Size

22.2.3.1.63 TXBRP Register (Offset = 2CCh) [reset = 0h]

TXBRP is shown in Figure 22-192 and described in Table 22-181.

Return to the Summary Table.

TXBRP

Figure 22-192. TXBRP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP																															
R-0h																															

Table 22-181. TXBRP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TRP	R	0h	Transmission Request Pending

22.2.3.1.64 TXBAR Register (Offset = 2D0h) [reset = 0h]

TXBAR is shown in Figure 22-193 and described in Table 22-182.

Return to the Summary Table.

TXBAR

Figure 22-193. TXBAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR																															
R/W0C-0h																															

Table 22-182. TXBAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	AR	R/W0C	0h	Add request

22.2.3.1.65 TXBCR Register (Offset = 2D4h) [reset = 0h]

TXBCR is shown in Figure 22-194 and described in Table 22-183.

Return to the Summary Table.

TXBCR

Figure 22-194. TXBCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR																															
R/W0C-0h																															

Table 22-183. TXBCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CR	R/W0C	0h	Cancellation Request

22.2.3.1.66 TXBTO Register (Offset = 2D8h) [reset = 0h]

TXBTO is shown in Figure 22-195 and described in Table 22-184.

Return to the Summary Table.

TXBTO

Figure 22-195. TXBTO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO																															
R-0h																															

Table 22-184. TXBTO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TO	R	0h	Transmission Occurred

22.2.3.1.67 TXBCF Register (Offset = 2DCh) [reset = 0h]

TXBCF is shown in Figure 22-196 and described in Table 22-185.

Return to the Summary Table.

TXBCF

Figure 22-196. TXBCF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF																															
R-0h																															

Table 22-185. TXBCF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CF	R	0h	Cancellation Finished

22.2.3.1.68 TXBTIE Register (Offset = 2E0h) [reset = 0h]

TXBTIE is shown in Figure 22-197 and described in Table 22-186.

Return to the Summary Table.

TXBTIE

Figure 22-197. TXBTIE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE																															
R/W-0h																															

Table 22-186. TXBTIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIE	R/W	0h	Transmission Interrupt Enable

22.2.3.1.69 TXBCIE Register (Offset = 2E4h) [reset = 0h]

TXBCIE is shown in Figure 22-198 and described in Table 22-187.

Return to the Summary Table.

TXBCIE

Figure 22-198. TXBCIE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE																															
R/W-0h																															

Table 22-187. TXBCIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CFIE	R/W	0h	Cancellation Finished Interrupt Enable

22.2.3.1.70 RES14 Register (Offset = 2E8h) [reset = 0h]

RES14 is shown in Figure 22-199 and described in Table 22-188.

Return to the Summary Table.

RES14

Figure 22-199. RES14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES14																															
R-0h																															

Table 22-188. RES14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES14	R	0h	Reserved

22.2.3.1.71 RES15 Register (Offset = 2ECh) [reset = 0h]

RES15 is shown in Figure 22-200 and described in Table 22-189.

Return to the Summary Table.

RES15

Figure 22-200. RES15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES15																															
R-0h																															

Table 22-189. RES15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES15	R	0h	Reserved

22.2.3.1.72 TXEFC Register (Offset = 2F0h) [reset = 0h]

TXEFC is shown in Figure 22-201 and described in Table 22-190.

Return to the Summary Table.

TXEFC

Figure 22-201. TXEFC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU68		EFWM						NU67		EFS					
R/W-0h		R/W-0h						R/W-0h		R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFSA														NU66	
R/W-0h														R/W-0h	

Table 22-190. TXEFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	NU68	R/W	0h	Reserved
29-24	EFWM	R/W	0h	Event FIFO Watermark
23-22	NU67	R/W	0h	Reserved
21-16	EFS	R/W	0h	Event FIFO Size
15-2	EFSA	R/W	0h	Event FIFO Start Address
1-0	NU66	R/W	0h	Reserved

22.2.3.1.73 TXEFS Register (Offset = 2F4h) [reset = 0h]

TXEFS is shown in Figure 22-202 and described in Table 22-191.

Return to the Summary Table.

TXEFS

Figure 22-202. TXEFS Register

31	30	29	28	27	26	25	24
NU72						TEFL	EFF
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
NU71			EFPI				
R-0h			R-0h				
15	14	13	12	11	10	9	8
NU70			EFGI				
R-0h			R-0h				
7	6	5	4	3	2	1	0
NU69		EFFL					
R-0h		R-0h					

Table 22-191. TXEFS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	NU72	R	0h	Reserved
25	TEFL	R	0h	Tx Event FIFO Element Lost
24	EFF	R	0h	Event FIFO Full
23-21	NU71	R	0h	Reserved
20-16	EFPI	R	0h	Event FIFO Put Index
15-13	NU70	R	0h	Reserved
12-8	EFGI	R	0h	Event FIFO Get Index
7-6	NU69	R	0h	Reserved
5-0	EFFL	R	0h	Event FIFO FIll Level

22.2.3.1.74 TXEFA Register (Offset = 2F8h) [reset = 0h]

TXEFA is shown in Figure 22-203 and described in Table 22-192.

Return to the Summary Table.

TXEFA

Figure 22-203. TXEFA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU73																											EFAI			
R-0h																											R-0h			

Table 22-192. TXEFA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	NU73	R	0h	Reserved
4-0	EFAI	R	0h	Event FIFO Acknowledge Index

22.2.3.1.75 RES16 Register (Offset = 2FCh) [reset = 0h]

RES16 is shown in Figure 22-204 and described in Table 22-193.

Return to the Summary Table.

RES16

Figure 22-204. RES16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES16																															
R-0h																															

Table 22-193. RES16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RES16	R	0h	Reserved

22.2.3.2 MSS_MCAN_ECC Registers

Table 22-194 lists the MSS_MCAN_ECC registers. All register offset addresses not listed in Table 22-194 should be considered as reserved locations and the register contents should not be modified.

Table 22-194. MSS_MCAN_ECC Registers

Offset	Acronym	Register Name	Section
0h	REV	REV	Section 22.2.3.2.1
8h	VECTOR	VECTOR	Section 22.2.3.2.2
Ch	STAT	STAT	Section 22.2.3.2.3
14h	CTRL	CTRL	Section 22.2.3.2.4
18h	ERR_CTRL1	ERR_CTRL1	Section 22.2.3.2.5
1Ch	ERR_CTRL2	ERR_CTRL2	Section 22.2.3.2.6
20h	ERR_STAT1	ERR_STAT1	Section 22.2.3.2.7
24h	ERR_STAT2	ERR_STAT2	Section 22.2.3.2.8
28h	ERR_STAT3	ERR_STAT3	Section 22.2.3.2.9
3Ch	SEC_EOI_REG	SEC_EOI_REG	Section 22.2.3.2.10
40h	SEC_STATUS_REG0	SEC_STATUS_REG0	Section 22.2.3.2.11
80h	SEC_ENABLE_SET_REG0	SEC_ENABLE_SET_REG0	Section 22.2.3.2.12
C0h	SEC_ENABLE_CLR_REG0	SEC_ENABLE_CLR_REG0	Section 22.2.3.2.13
13Ch	DED_EOI_REG	DED_EOI_REG	Section 22.2.3.2.14
140h	DED_STATUS_REG0	DED_STATUS_REG0	Section 22.2.3.2.15
180h	DED_ENABLE_SET_REG0	DED_ENABLE_SET_REG0	Section 22.2.3.2.16
1C0h	DED_ENABLE_CLR_REG0	DED_ENABLE_CLR_REG0	Section 22.2.3.2.17
200h	AGGR_ENABLE_SET	AGGR_ENABLE_SET	Section 22.2.3.2.18
204h	AGGR_ENABLE_CLR	AGGR_ENABLE_CLR	Section 22.2.3.2.19
208h	AGGR_STATUS_SET	AGGR_STATUS_SET	Section 22.2.3.2.20
20Ch	AGGR_STATUS_CLR	AGGR_STATUS_CLR	Section 22.2.3.2.21

22.2.3.2.1 REV Register (Offset = 0h) [reset = 66A0EA00h]

REV is shown in Figure 22-205 and described in Table 22-195.

Return to the Summary Table.

Aggregator Revision Register

Figure 22-205. REV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		BU		MODULE_ID											
R-1h		R-2h		R-6A0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVRTL					REVMAJ			CUSTOM		REVMIN					
R-1Dh					R-2h			R-0h		R-0h					

Table 22-195. REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Scheme
29-28	BU	R	2h	bu
27-16	MODULE_ID	R	6A0h	Module ID
15-11	REVRTL	R	1Dh	RTL version
10-8	REVMAJ	R	2h	Major version
7-6	CUSTOM	R	0h	Custom version
5-0	REVMIN	R	0h	Minor version

22.2.3.2.2 VECTOR Register (Offset = 8h) [reset = 0h]

VECTOR is shown in Figure 22-206 and described in Table 22-196.

Return to the Summary Table.

ECC Vector Register

Figure 22-206. VECTOR Register

31	30	29	28	27	26	25	24
NU1							RD_SVBUS_DONE
R-0h							R/W-0h
23	22	21	20	19	18	17	16
RD_SVBUS_ADDR							
R/W-0h							
15	14	13	12	11	10	9	8
RD_SVBUS	NU0					ECC_VEC	
R/W-0h	R-0h					R/W-0h	
7	6	5	4	3	2	1	0
ECC_VEC							
R/W-0h							

Table 22-196. VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU1	R	0h	Reserved
24	RD_SVBUS_DONE	R/W	0h	Status to indicate if read on serial VBUS is complete, write of any value will clear this bit. Writing 1 to any bit will clear the corresponding bits. Reads do not alter the value of the field.
23-16	RD_SVBUS_ADDR	R/W	0h	Read address
15	RD_SVBUS	R/W	0h	Write 1 to trigger a read on the serial VBUS. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field.
14-11	NU0	R	0h	Reserved
10-0	ECC_VEC	R/W	0h	Value written to select the corresponding ECC RAM for control or status

22.2.3.2.3 STAT Register (Offset = Ch) [reset = 2h]

STAT is shown in Figure 22-207 and described in Table 22-197.

Return to the Summary Table.

Misc Status

Figure 22-207. STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2																					NUM_RAMS										
R-0h																					R-2h										

Table 22-197. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU2	R	0h	Reserved
10-0	NUM_RAMS	R	2h	Indicates the number of RAMS serviced by the ECC aggregator

22.2.3.2.4 CTRL Register (Offset = 14h) [reset = 187h]

CTRL is shown in Figure 22-208 and described in Table 22-198.

Return to the Summary Table.

CTRL

Figure 22-208. CTRL Register

31	30	29	28	27	26	25	24
NU3							
R-0h							
23	22	21	20	19	18	17	16
NU3							
R-0h							
15	14	13	12	11	10	9	8
NU3							CHECK TIMEOUT
R-0h							W-1h
7	6	5	4	3	2	1	0
CHECK PARITY	ERROR_ONCE	FORCE_N_ROW	FORCE_DED	FORCE_SEC	EN_RMW	ECC_CHK	ECC_EN
W-1h	W-0h	W-0h	W-0h	W-0h	W-1h	W-1h	W-1h

Table 22-198. CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	NU3	R	0h	TI Internal : Reserved
8	CHECK TIMEOUT	W	1h	TI Internal : Check timeout
7	CHECK PARITY	W	1h	TI Internal : Check Parity
6	ERROR_ONCE	W	0h	TI Internal : Force Error only once
5	FORCE_N_ROW	W	0h	TI Internal : Force Error on any RAM read
4	FORCE_DED	W	0h	TI Internal : Force Double Bit Error
3	FORCE_SEC	W	0h	TI Internal : Force Single Bit Error
2	EN_RMW	W	1h	TI Internal : Enable rmw
1	ECC_CHK	W	1h	TI Internal : Enable ECC check
0	ECC_EN	W	1h	TI Internal : Enable ECC

22.2.3.2.5 ERR_CTRL1 Register (Offset = 18h) [reset = 0h]

ERR_CTRL1 is shown in Figure 22-209 and described in Table 22-199.

Return to the Summary Table.

ERR_CTRL1

Figure 22-209. ERR_CTRL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_ROW																															
R/W-0h																															

Table 22-199. ERR_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECC_ROW	R/W	0h	TI Internal : Row address where single or double-bit error needs to be applied. This is ignored if force_n_row is set

22.2.3.2.6 ERR_CTRL2 Register (Offset = 1Ch) [reset = 0h]

ERR_CTRL2 is shown in Figure 22-210 and described in Table 22-200.

Return to the Summary Table.

ERR_CTRL2

Figure 22-210. ERR_CTRL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BIT2																ECC_BIT1															
R/W-0h																R/W-0h															

Table 22-200. ERR_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT2	R/W	0h	TI Internal : Data bit that needs to be flipped if double bit error needs to be forced
15-0	ECC_BIT1	R/W	0h	TI Internal : Data bit that needs to be flipped when force_sec is set

22.2.3.2.7 ERR_STAT1 Register (Offset = 20h) [reset = 0h]

ERR_STAT1 is shown in Figure 22-211 and described in Table 22-201.

Return to the Summary Table.

ERR_STAT1

Figure 22-211. ERR_STAT1 Register

31	30	29	28	27	26	25	24
ECC_BIT1_STS							
R-0h							
23	22	21	20	19	18	17	16
ECC_BIT1_STS							
R-0h							
15	14	13	12	11	10	9	8
CLR_ECC_CTRL_REG	CLR_ECC_PAR		CLR_ECC_OTHER	CLR_ECC_DED		CLR_ECC_SEC	
W-0h	W-0h		W-0h	W-0h		W-0h	
7	6	5	4	3	2	1	0
ECC_CTRL_REG	ECC_PAR		ECC_OTHER	ECC_DED		ECC_SEC	
W-0h	W-0h		W-0h	W-0h		W-0h	

Table 22-201. ERR_STAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT1_STS	R	0h	TI Internal : Data bit that corresponds to the single-bit error
15	CLR_ECC_CTRL_REG	W	0h	TI Internal : Clear Ctrl Reg Error Status. Write 1 to clear. This bit is self clearing.
14-13	CLR_ECC_PAR	W	0h	TI Internal : Clear Parity Error Status. Write 1 to clear. This bit is self clearing.
12	CLR_ECC_OTHER	W	0h	TI Internal : Clear Other Error Status. Write 1 to clear. This bit is self clearing.
11-10	CLR_ECC_DED	W	0h	TI Internal : Clear Double Bit Error Status. Write 1 to clear. This bit is self clearing.
9-8	CLR_ECC_SEC	W	0h	TI Internal : Clear Single Bit Error Status. Write 1 to clear. This bit is self clearing.
7	ECC_CTRL_REG	W	0h	TI Internal : Force ctrl reg pending interrupt. Write 1 to set. This bit is self clearing.
6-5	ECC_PAR	W	0h	TI Internal : Force ECC parity pending interrupt. Write 1 to set. This bit is self clearing.
4	ECC_OTHER	W	0h	TI Internal : Force ECC other pending interrupt. Write 1 to set. This bit is self clearing.
3-2	ECC_DED	W	0h	TI Internal : Force ECC DED pending interrupt. Write 1 to set. This bit is self clearing.
1-0	ECC_SEC	W	0h	TI Internal : Force ECC SEC pending interrupt. Write 1 to set. This bit is self clearing.

22.2.3.2.8 ERR_STAT2 Register (Offset = 24h) [reset = 0h]

ERR_STAT2 is shown in Figure 22-212 and described in Table 22-202.

Return to the Summary Table.

ERR_STAT2

Figure 22-212. ERR_STAT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_ROW																															
R-0h																															

Table 22-202. ERR_STAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECC_ROW	R	0h	TI Internal : Row address where the single or double-bit error has occurred

22.2.3.2.9 ERR_STAT3 Register (Offset = 28h) [reset = 0h]

ERR_STAT3 is shown in Figure 22-213 and described in Table 22-203.

Return to the Summary Table.

ERR_STAT3

Figure 22-213. ERR_STAT3 Register

31	30	29	28	27	26	25	24
NU6							
R-0h							
23	22	21	20	19	18	17	16
NU6							
R-0h							
15	14	13	12	11	10	9	8
NU6						CLR_TIMEOUT_PEND	NU5
R-0h						W-0h	R-0h
7	6	5	4	3	2	1	0
NU5						TIMEOUT_PEND	NU4
R-0h						W-0h	R-0h

Table 22-203. ERR_STAT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU6	R	0h	TI Internal : Reserved
9	CLR_TIMEOUT_PEND	W	0h	TI Internal : Clear timeout pending
8-2	NU5	R	0h	TI Internal : Reserved
1	TIMEOUT_PEND	W	0h	TI Internal : Timeout pending
0	NU4	R	0h	TI Internal : Reserved

22.2.3.2.10 SEC_EOI_REG Register (Offset = 3Ch) [reset = 0h]

SEC_EOI_REG is shown in Figure 22-214 and described in Table 22-204.

Return to the Summary Table.

EOI Register

Figure 22-214. SEC_EOI_REG Register

31	30	29	28	27	26	25	24
NU7							
R-0h							
23	22	21	20	19	18	17	16
NU7							
R-0h							
15	14	13	12	11	10	9	8
NU7							
R-0h							
7	6	5	4	3	2	1	0
NU7							SEC_EOI_WR
R-0h							R/W-0h

Table 22-204. SEC_EOI_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU7	R	0h	Reserved
0	SEC_EOI_WR	R/W	0h	EOI Register. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field. This bit is self clearing, reading this bit will return 0.

22.2.3.2.11 SEC_STATUS_REG0 Register (Offset = 40h) [reset = 0h]

SEC_STATUS_REG0 is shown in Figure 22-215 and described in Table 22-205.

Return to the Summary Table.

Interrupt Status Register 0

Figure 22-215. SEC_STATUS_REG0 Register

31	30	29	28	27	26	25	24
NU8							
R-0h							
23	22	21	20	19	18	17	16
NU8							
R-0h							
15	14	13	12	11	10	9	8
NU8							
R-0h							
7	6	5	4	3	2	1	0
NU8						CTRL_EDC_VBUSS_PEND	SEC_PEND
R-0h						R-0h	R-0h

Table 22-205. SEC_STATUS_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU8	R	0h	Reserved
1	CTRL_EDC_VBUSS_PEND	R	0h	Interrupt Pending Status for ctrl_edc_vbuss_pend.
0	SEC_PEND	R	0h	Interrupt Pending Status for msgmem_pend.

22.2.3.2.12 SEC_ENABLE_SET_REG0 Register (Offset = 80h) [reset = 0h]

SEC_ENABLE_SET_REG0 is shown in Figure 22-216 and described in Table 22-206.

Return to the Summary Table.

Interrupt Enable Set Register 0

Figure 22-216. SEC_ENABLE_SET_REG0 Register

31	30	29	28	27	26	25	24
NU9							
R-0h							
23	22	21	20	19	18	17	16
NU9							
R-0h							
15	14	13	12	11	10	9	8
NU9							
R-0h							
7	6	5	4	3	2	1	0
NU9						CTRL_EDC_VBUSS_ENABLE_SET	SEC_EN_SET
R-0h						R/W-0h	R/W-0h

Table 22-206. SEC_ENABLE_SET_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU9	R	0h	Reserved
1	CTRL_EDC_VBUSS_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for ctrl_edc_vbuss_pend. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field.
0	SEC_EN_SET	R/W	0h	Interrupt Enable Set Register for msgmem_pend. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field.

22.2.3.2.13 SEC_ENABLE_CLR_REG0 Register (Offset = C0h) [reset = 0h]

SEC_ENABLE_CLR_REG0 is shown in Figure 22-217 and described in Table 22-207.

Return to the Summary Table.

Interrupt Enable Clear Register 0

Figure 22-217. SEC_ENABLE_CLR_REG0 Register

31	30	29	28	27	26	25	24
NU10							
R-0h							
23	22	21	20	19	18	17	16
NU10							
R-0h							
15	14	13	12	11	10	9	8
NU10							
R-0h							
7	6	5	4	3	2	1	0
NU10						CTRL_EDC_VBUSS_ENABLE_CLR	SEC_EN_CLR
R-0h						R/W-0h	R/W-0h

Table 22-207. SEC_ENABLE_CLR_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU10	R	0h	Reserved
1	CTRL_EDC_VBUSS_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for ctrl_edc_vbuss_pend. Writing 1 to any bit will clear the corresponding bits. Reads do not alter the value of the field. Reading this bit will return 0.
0	SEC_EN_CLR	R/W	0h	Interrupt Enable Clear Register for msgmem_pend. Writing 1 to any bit will clear the corresponding bits. Reads do not alter the value of the field. Reading this bit will return 0.

22.2.3.2.14 DED_EOI_REG Register (Offset = 13Ch) [reset = 0h]

DED_EOI_REG is shown in Figure 22-218 and described in Table 22-208.

Return to the Summary Table.

EOI Register

Figure 22-218. DED_EOI_REG Register

31	30	29	28	27	26	25	24
NU11							
R-0h							
23	22	21	20	19	18	17	16
NU11							
R-0h							
15	14	13	12	11	10	9	8
NU11							
R-0h							
7	6	5	4	3	2	1	0
NU11							DED_EOI_WR
R-0h							R/W-0h

Table 22-208. DED_EOI_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU11	R	0h	Reserved
0	DED_EOI_WR	R/W	0h	EOI Register. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field. This bit is self clearing, reading this bit will return 0.

22.2.3.2.15 DED_STATUS_REG0 Register (Offset = 140h) [reset = 0h]

DED_STATUS_REG0 is shown in Figure 22-219 and described in Table 22-209.

Return to the Summary Table.

Interrupt Status Register 0

Figure 22-219. DED_STATUS_REG0 Register

31	30	29	28	27	26	25	24
NU12							
R-0h							
23	22	21	20	19	18	17	16
NU12							
R-0h							
15	14	13	12	11	10	9	8
NU12							
R-0h							
7	6	5	4	3	2	1	0
NU12						CTRL_EDC_VBUSS_PEND	DED_PEND
R-0h						R-0h	R-0h

Table 22-209. DED_STATUS_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU12	R	0h	Reserved
1	CTRL_EDC_VBUSS_PEND	R	0h	Interrupt Pending Status for ctrl_edc_vbuss_pend.
0	DED_PEND	R	0h	Interrupt Pending Status for msgmem_pend.

22.2.3.2.16 DED_ENABLE_SET_REG0 Register (Offset = 180h) [reset = 0h]

DED_ENABLE_SET_REG0 is shown in Figure 22-220 and described in Table 22-210.

Return to the Summary Table.

Interrupt Enable Set Register 0

Figure 22-220. DED_ENABLE_SET_REG0 Register

31	30	29	28	27	26	25	24
NU13							
R-0h							
23	22	21	20	19	18	17	16
NU13							
R-0h							
15	14	13	12	11	10	9	8
NU13							
R-0h							
7	6	5	4	3	2	1	0
NU13						CTRL_EDC_VBUSS_ENABLE_SET	DED_EN_SET
R-0h						R/W-0h	R/W-0h

Table 22-210. DED_ENABLE_SET_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU13	R	0h	Reserved
1	CTRL_EDC_VBUSS_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for ctrl_edc_vbuss_pend. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field.
0	DED_EN_SET	R/W	0h	Interrupt Enable Set Register for msgmem_pend. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field.

22.2.3.2.17 DED_ENABLE_CLR_REG0 Register (Offset = 1C0h) [reset = 0h]

DED_ENABLE_CLR_REG0 is shown in Figure 22-221 and described in Table 22-211.

Return to the Summary Table.

Interrupt Enable Clear Register 0

Figure 22-221. DED_ENABLE_CLR_REG0 Register

31	30	29	28	27	26	25	24
NU14							
R-0h							
23	22	21	20	19	18	17	16
NU14							
R-0h							
15	14	13	12	11	10	9	8
NU14							
R-0h							
7	6	5	4	3	2	1	0
NU14						CTRL_EDC_VBUSS_ENABLE_CLR	DED_EN_CLR
R-0h						R/W-0h	R/W-0h

Table 22-211. DED_ENABLE_CLR_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU14	R	0h	Reserved
1	CTRL_EDC_VBUSS_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for ctrl_edc_vbuss_pend. Writing 1 to any bit will clear the corresponding bits. Reads do not alter the value of the field. Reading this bit will return 0.
0	DED_EN_CLR	R/W	0h	Interrupt Enable Clear Register for msgmem_pend. Writing 1 to any bit will clear the corresponding bits. Reads do not alter the value of the field. Reading this bit will return 0.

22.2.3.2.18 AGGR_ENABLE_SET Register (Offset = 200h) [reset = 0h]

AGGR_ENABLE_SET is shown in Figure 22-222 and described in Table 22-212.

Return to the Summary Table.

AGGR interrupt enable set Register

Figure 22-222. AGGR_ENABLE_SET Register

31	30	29	28	27	26	25	24
NU15							
R-0h							
23	22	21	20	19	18	17	16
NU15							
R-0h							
15	14	13	12	11	10	9	8
NU15							
R-0h							
7	6	5	4	3	2	1	0
NU15						TIMEOUT	PARITY
R-0h						R/W-0h	R/W-0h

Table 22-212. AGGR_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU15	R	0h	Reserved
1	TIMEOUT	R/W	0h	Interrupt Enable Set Register for svbus timeout errors. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field.
0	PARITY	R/W	0h	Interrupt Enable Set Register for parity errors. Writing 1 to any bit will set the corresponding bit. Reads do not alter the value of the field.

22.2.3.2.19 AGGR_ENABLE_CLR Register (Offset = 204h) [reset = 0h]

AGGR_ENABLE_CLR is shown in Figure 22-223 and described in Table 22-213.

Return to the Summary Table.

AGGR interrupt enable clear Register

Figure 22-223. AGGR_ENABLE_CLR Register

31	30	29	28	27	26	25	24
NU16							
R-0h							
23	22	21	20	19	18	17	16
NU16							
R-0h							
15	14	13	12	11	10	9	8
NU16							
R-0h							
7	6	5	4	3	2	1	0
NU16						TIMEOUT	PARITY
R-0h						R/W-0h	R/W-0h

Table 22-213. AGGR_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU16	R	0h	Reserved
1	TIMEOUT	R/W	0h	Interrupt Enable Clear for svbus timeout errors. Writing 1 to any bit will clear the corresponding bits. Reads do not alter the value of the field. Reading this bit will return 0.
0	PARITY	R/W	0h	Interrupt Enable Clear for parity errors. Writing 1 to any bit will clear the corresponding bits. Reads do not alter the value of the field. Reading this bit will return 0.

22.2.3.2.20 AGGR_STATUS_SET Register (Offset = 208h) [reset = 0h]

AGGR_STATUS_SET is shown in Figure 22-224 and described in Table 22-214.

Return to the Summary Table.

AGGR interrupt status set Register

Figure 22-224. AGGR_STATUS_SET Register

31	30	29	28	27	26	25	24
NU17							
R-0h							
23	22	21	20	19	18	17	16
NU17							
R-0h							
15	14	13	12	11	10	9	8
NU17							
R-0h							
7	6	5	4	3	2	1	0
NU17				TIMEOUT		PARITY	
R-0h				R/W-0h		R/W-0h	

Table 22-214. AGGR_STATUS_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU17	R	0h	Reserved
3-2	TIMEOUT	R/W	0h	Interrupt status set for svbus timeout errors. A write to increment field. Writing a value to this field increment the field value by the value written. Reads do not alter the value of the field.
1-0	PARITY	R/W	0h	Interrupt status set for parity errors. A write to increment field. Writing a value to this field increment the field value by the value written. Reads do not alter the value of the field.

22.2.3.2.21 AGGR_STATUS_CLR Register (Offset = 20Ch) [reset = 0h]

AGGR_STATUS_CLR is shown in Figure 22-225 and described in Table 22-215.

Return to the Summary Table.

AGGR interrupt status clear Register

Figure 22-225. AGGR_STATUS_CLR Register

31	30	29	28	27	26	25	24
NU18							
R-0h							
23	22	21	20	19	18	17	16
NU18							
R-0h							
15	14	13	12	11	10	9	8
NU18							
R-0h							
7	6	5	4	3	2	1	0
NU18				TIMEOUT		PARITY	
R-0h				R/W-0h		R/W-0h	

Table 22-215. AGGR_STATUS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU18	R	0h	Reserved
3-2	TIMEOUT	R/W	0h	Interrupt status clear for svbus timeout errors. A write to decrement field. Writing a value to this field decrements the field value by the value written. Reads do not alter the value of the field.
1-0	PARITY	R/W	0h	Interrupt status clear for parity errors. A write to decrement field. Writing a value to this field decrements the field value by the value written. Reads do not alter the value of the field.

Multi-Buffered Serial Peripheral Interface Module (MibSPI)

This section provides the specifications for a 16-bit configurable synchronous multi-buffered multi-pin serial peripheral interface (MibSPI). This section also provides the specifications for MibSPI with Parallel Pin Option (MibSPIP). The MibSPI is a programmable-length shift register used for high-speed communication between external peripherals or other microcontrollers.

Throughout this section, all references to SPI also apply to MibSPI/MibSPIP, unless otherwise noted.

Table 23-1. Register Control From MSS-RCM Configuration Space

spia_trig_src[1:0]	Driven for configuration register from MSS_RCM	MSS_RCM:SPITRIGSRC: SPIATRIG1 MSS_RCM:SPITRIGSRC: SPIATRIG0
spia_mem_init	Driven for configuration register from MSS_RCM	MSS_RCM:MEMINITSTART: SPIAMEM
spia_mem_init_done	Status, which can be read from MSS_RCM Register space	MSS_RCM:MEMINITDONE: SPIAMEM

Table 23-2. Register Control From MSS-RCM Configuration Space

spib_trig_src[5:0]	Driven for configuration register from MSS_RCM	MSS_RCM:SPITRIGSRC: SPIBTRIG[5:0]
spib_mem_init	Driven for configuration register from MSS_RCM	MSS_RCM:MEMINITSTART: SPIBMEM
spib_mem_init_done	Status, which can be read from MSS_RCM Register space	MSS_RCM:MEMINITDONE: SPIBMEM

NOTE: This section describes a superset implementation of the MibSPI/SPI modules that includes features and functionality that may not be available on some devices. Device-specific content that should be determined by referencing the datasheet includes DMA functionality, MibSPI RAM size, number of transfer groups, number of chip selects, parallel mode support, and availability of 5-pin operation (SPInENA).

Topic **Page**

23.1 Overview **2872**
23.2 Basic Operation **2874**
23.3 Control Registers **2909**
23.4 Multi-Buffer RAM **3142**
23.5 Parity\ECC Memory **3148**
23.6 MibSPI Pin Timing Parameters **3153**

23.1 Overview

23.1.1 Features

The MibSPI/SPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (two to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The MibSPI/SPI is normally used for communication between the microcontroller and external peripherals or another microcontroller. Typical applications include interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, and analog-to-digital converters. MibSPI is an Extension of SPI. MibSPI works in 2 Modes.

- Compatibility Mode
- Multi-buffer Mode

The Compatibility mode of MibSPI makes it behave exactly like that of SPI and ensures full compatibility with the same. Everything described about compatibility mode of MibSPI , in this document, is directly applicable to SPI.

The Multi-buffer mode of operation is specific to MibSPI alone. This feature is not available in SPI.

The MibSPI supports memory fault detection/correction via internal Parity/ECC circuit. MibSPI is configurable to include or not include Memory Parity/ECC logic during circuit synthesis.

The SPI / MibSPI can be configured in three pin, four pin or five pin mode of operation. The SPI / MibSPI allows multiple programmable chip-selects.

The MibSPI has a programmable Multi-buffer array that enables programed transmission to be completed without CPU intervention. The buffers are combined in different transfer groups that could be triggered by external events (Timers, I/O, and so on) or by the internal tick counter. The internal tick counter can support periodic trigger events. Each buffer of the MibSPI can be associated with different DMA channels in different transfer group, allowing the user to move data from/to internal memory to/from external slave with a minimal CPU interaction.

The pins SPICLK, SPISIMO and SPISOMI are used in all MibSPI pin modes. The pins SPIENA and SPISCS[7:0] are optional and may be used if the pin are present on a given device.

The SPI has the following attributes:

- 16-bit shift register
- Receive buffer register
- 8-bit baud clock generator
- Serial Clock (SPICLK) I/O pin
- SPI enable ($\overline{\text{SPIENA}}$) pin (4 or 5-pin mode only)
- Up to 8 slave chip select (Slave Chip Select $\overline{\text{SPISCS}}$[7:0]) pin (4 or 5-pin mode only)
- SPI pins can be used as functional or digital Input/Output pins (GIOs)

The SPI / MibSPI allows software to program the following options:

- SPISOMI/SPISIMO pin direction configuration
- SPICLK pin source (external/internal)
- MibSPI pins as functional or digital I/O pins. For each Buffer, following features can be selected from 4 different combinations of Formats using the control fields in the buffer.
  - SPICLK frequency
  - Charecter length
  - Phase,Polarity
  - Enable/Disable parity for transmit and receive
  - Enable/Disable timers for ChipSelect Hold and Setup timers
  - Direction of shifting, MSBit first or LSBit first
  - Configurable Parallel modes to use multiple SIMO/SOMI pin
  - Configurable number of ChipSelects

In Multi-buffer Mode, in addition to the above, many other features are configurable

- Number of buffers for each peripheral (or data source/destination, up to 256 buffers supported) or group (up to 8 groupings)
- Number of DMA controlled buffers and number of DMA request channels (up to 8 for each of transmit and receive)
- Triggers for each groups, trigger types, trigger sources for individual groups(up to 14 external trigger sources and 1 internal trigger source)
- Number of DMA transfers for each buffer (up to 65536 for up to 8 buffers)
- Un-interrupted DMA buffer transfer (NOBREAK buffer)

NOTE: SIMO - Slave In Master Out Pin
SOMI - Slave Out Master In Pin
CS - SPI Chip Select Pin
ENA - SPI Enable Pin.

23.1.2 Pin Configurations

The SPI supports data connections as shown in Table 23-3.

Table 23-3. Pin Configurations

Pin	Master Mode		Slave Mode	
SPICLK	Drives the clock to external devices		Receives the clock from the external master	
SPISOMI	Receives data from the external slave		Sends data to the external master	
SPISIMO	Transmits data to the external slave		Receives data from the external master	
SPIENA	**SPIENA disabled:** GIO	**SPIENA enabled:** Receives ENA signal from the external slave	**SPIENA disabled:** GIO	**SPIENA enabled:** Receives ENA signal from the external master
SPICS[7:0]	**SPICS disabled:** GIO	**SPICS enabled:** Selects one or more slave devices	**SPICS disabled:** GIO	**SPICS enabled:** Receives the CS signal from the external master

NOTE:

1. When the SPICS[3:0] signals are disabled, the chip-select field in the transmit data is not used.
2. When the SPIENA signal is disabled, the $\overline{\text{SPIENA}}$ pin is ignored in master mode, and not driven as part of the SPI transaction in slave mode.

23.1.3 MibSPI /SPI Configurations

Table 23-4. MibSPI/SPI Configurations

MibSPIx/SPIx	I/Os
MIBSPI1 (MIBSPI-A)	MIBSPI1SIMO[1:0], MIBSPI1SOMI[1:0], MIBSPI1CLK, MIBSPI1nCS[5:0], MIBSPI1nENA
MIBSPI2 (MIBSPI-B)	MIBSPI2SIMO[1:0], MIBSPI2SOMI[1:0], MIBSPI2CLK, MIBSPI2nCS[5:0], MIBSPI2nENA
MIBSPI3 (MIBSPI-C)	MIBSPI3SIMO[1:0], MIBSPI3SOMI[1:0], MIBSPI3CLK, MIBSPI3nCS[5:0], MIBSPI3nENA
MIBSPI4 (MIBSPI-D)	MIBSPI4SIMO[1:0], MIBSPI4SOMI[1:0], MIBSPI4CLK, MIBSPI4nCS[5:0], MIBSPI4nENA
MIBSPI5 (MIBSPI-E)	MIBSPI5SIMO[1:0], MIBSPI5SOMI[1:0], MIBSPI51CLK, MIBSPI5nCS[5:0], MIBSP5nENA
SPI1 (SPI-A)	SPI1SIMO, ZSPI1SOMI, SPI1CLK, SPI2nCS[1:0], SPI1nENA
SPI2 (SPI-B)	SPI2SIMO, ZSPI2SOMI, SPI2CLK, SPI2nCS[1:0], SPI2nENA
SPI3 (SPI-C)	SPI3SIMO, ZSPI3SOMI, SPI3CLK, SPI3nCS[1:0], SPI3nENA

23.2 Basic Operation

This section details the basic operation principle of the SPI mode and the MibSPI mode operation of the device.

23.2.1 SPI Mode

The SPI can be configured via software to operate as either a master or a slave. The MASTER bit (SPIGCR1[0]) selects the configuration of the SPISIMO and SPISOMI pins. CLKMOD bit (SPIGCR1[1]) determines whether an internal or external clock source will be used.

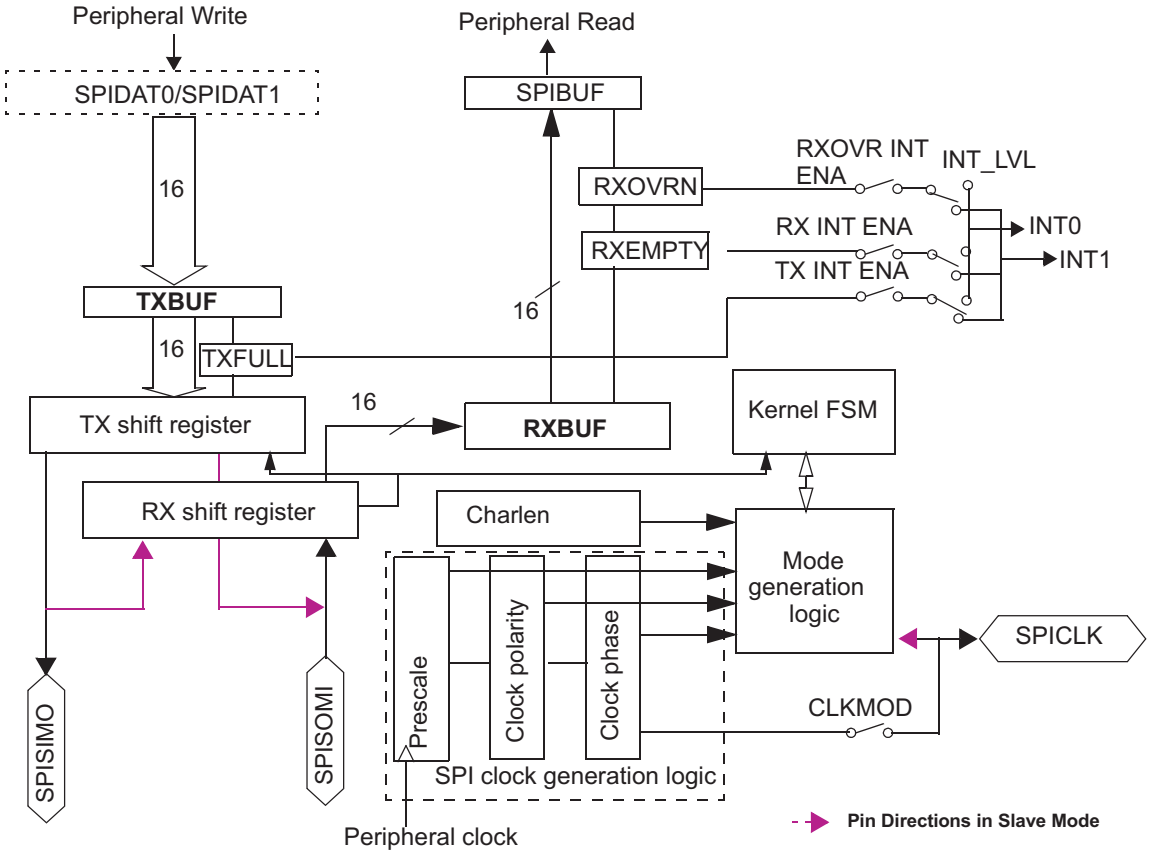
The slave chip select ($\overline{\text{SPISCS}}$[7:0]) pins, are used when communicating with multiple slave devices. When the a write occurs to SPIDAT1 in master mode, the $\overline{\text{SPISCS}}$ pins are automatically driven to select the specified slave.

Handshaking mechanism, provided by the $\overline{\text{SPIENA}}$ pin, enables a slave SPI to delay the generation of the clock signal supplied by the master if it is not prepared for the next exch ange of data.

23.2.1.1 SPI Mode Operation Block Diagram

Figure 23-1 shows the SPI transaction hardware. TXBUF and RXBUF are internal buffers that are intended to improve the overall throughput of data transfer. TXBUF is a transmit buffer, while RXBUF is a receive buffer.

Figure 23-1. SPI Functional Logic Diagram



1 This is a representative diagram, which shows three-pin mode hardware.
2 TXBUF, RXBUF, and SHIFT_REGISTER are user-invisible registers.
3 SPIDAT0 and SPIDAT1 are user-visible, and are physically mapped to the contents of TXBUF.
4 SPISIMO, SPISOMI, SPICLK pin directions depend on the Master or Slave Mode.

23.2.1.2 Data Flow and Handling for TX and RX

23.2.1.2.1 Data Sequencing when SPIDAT0 or SPIDAT1 is Written

- If both the TX shift register and TXBUF are empty, then the data is directly copied to the TX shift register. For devices with DMA, if DMA is enabled, a transmit DMA request (TX_DMA_REQ) is generated to cause the next word to be fetched. If transmit interrupts are enabled, a transmitter-empty interrupt is generated.
- If the TX shift register is already full or is in the process of shifting and if TXBUF is expty then the data written to SPIDAT0 / SPIDAT1 is copied to TXBUF and TXFULL flag is set to 1 at the same time.
- When a shift operation is complete, data from the TXBUF (if it is full) is copied into TX shift register and the TXFULL flag is cleared to 0 to indicate that next data can be fetched. A transmit DMA request (if enabled) or a transmitter-empty interrupt (if enabled) is generated at the same time.

23.2.1.2.2 Data Sequencing when All Bits Shifted into RXSHIFT Register

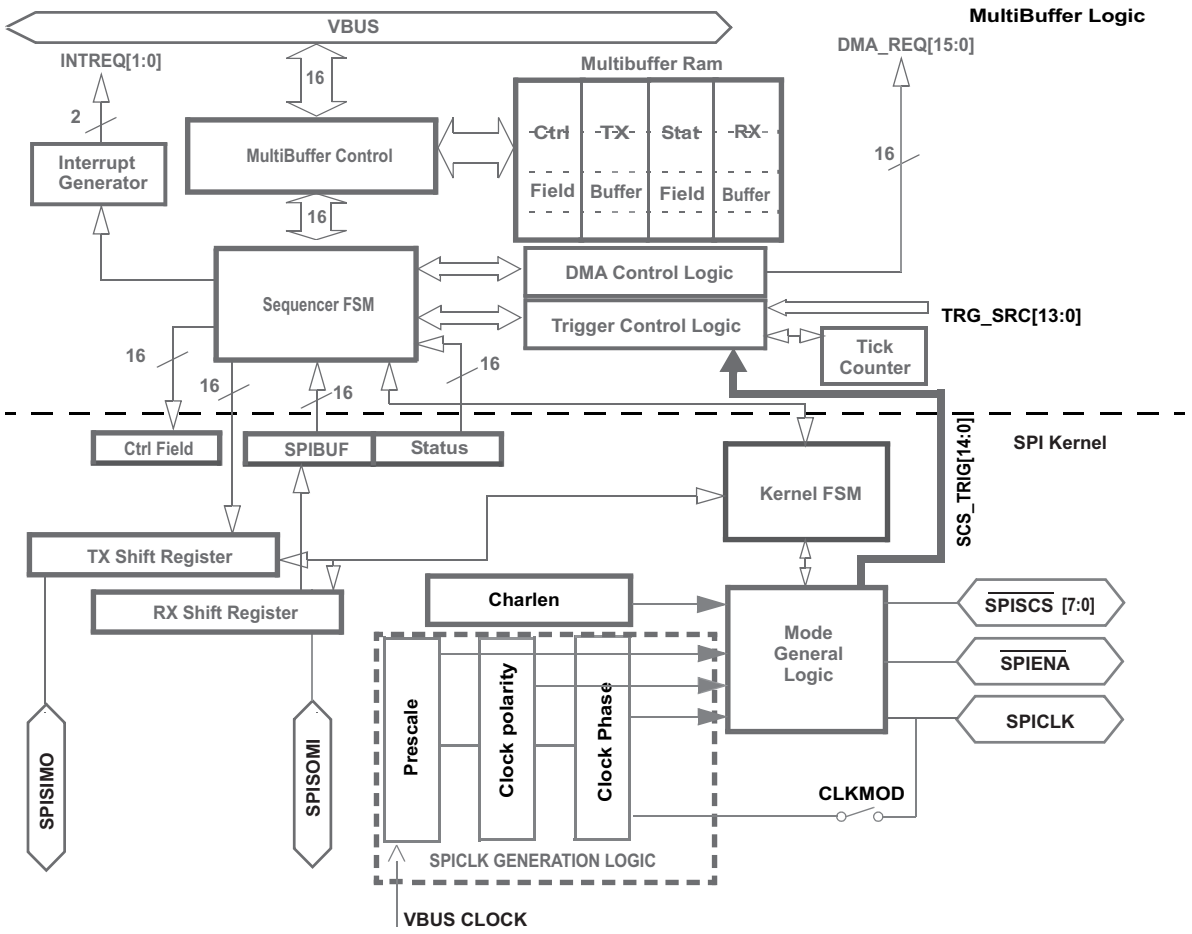
- If both SPIBUF and RXBUF are empty, the received data in RX shift register is directly copied into SPIBUF and the receive DMA request (if enabled) is generated and the receive-interrupt (if enabled) is generated. The RXEMPTY flag in SPIBUF is cleared at the same time.
- If SPIBUF is already full at the end of receive completion, the RX shift register contents is copied to RXBUF. A receive DMA request is generated, if enabled. The receive complete interrupt line remains high.
- If SPIBUF is read by the CPU or DMA and if RXBUF is full, then the contents of RXBUF are copied to SPIBUF as soon as SPIBUF is read. RXEMPTY flag remains cleared, indicating that SPIBUF is still full.
- If both SPIBUF and RXBUF are full, then RXBUF will be overwritten and the RXOVR interrupt flag is set and an interrupt is generated, if enabled.

NOTE: Prefetching is done only in Master mode. In Slave mode, since the TG to be serviced is known only after a valid ChipSelect assertion, no prefetching is done.

23.2.2 MibSPI Mode

Figure 23-2 shows multi-buffered mode operation. In Multi-buffer mode the transmit data has to be written to the TXRAM locations and the receive data has to be read from RXRAM locations of the multi-buffer RAM. A MibSPI supports up to 256 locations each for Transmit and Receive Data.

Figure 23-2. MibSPI Functional Logic Diagram



23.2.2.1 Data Handling for TX and RX Transfer Groups

23.2.2.1.1 Data Sequencing of a Transmit Data

In multi-buffer mode, any buffer that needs to be transmitted over by the SPI, should be associated with a Transfer Group. Each TG (Transfer Group) will have a Trigger Source based on which it'll be triggered. Once a TG is triggered, the buffers belonging to it will be transmitted.

Sequencer (FSM) controls the data flow from the multi-buffer RAM to the Shift Register. The Multi-buffer Control Logic has arbitration logic between VBUS and the Sequencer accessing the multi-buffer RAM. Sequencer picks up a highest priority Transfer Group from among the active TGs to be serviced. For the selected TG the starting buffer to be transferred is obtained from the PSTART of the respective TGxCTRL register.

Sequencer requests for the selected buffer through the Multi-buffer Control Logic, and once it receives the data, it reads the control fields to determine the subsequent action. Once the buffer is determined to be ready for transfer, the data is written to the TX SHIFT REGISTER by the Sequencer. This triggers the Kernel FSM to initiate the SPI transfer.

Once the Sequencer is finished writing to the TX SHIFT REGISTER, it prefetches the next buffer to be transferred from the multi-buffer RAM and stores the Data.

Once the Sequencer is finished writing to the TX SHIFT REGISTER, it prefetches the next buffer to be transferred from the multi-buffer RAM and stores the Data.

Sequencer writes the prefetched Transmit Data to the Shift Register immediately upon request by the Kernel. This way, the throughput of the SPI transfer is increased in Master mode of operation. In case of Slave mode, after the Receive data is copied to the RX RAM, Sequencer waits for the next active Chip Select trigger to fetch the next data.

23.2.2.1.2 Data Sequencing of the Received Data

At the end of a SPI transfer, the received Data is copied to SPIBUF register and then forwarded to the Sequencer. The Sequencer then, requests the Multi-buffer Control Logic to write the received data to the respective RXRAM location. Along with Received Data, the Status fields like Transmission Error Flags and the Last Chip Select Number (LCSNR) are forwarded to be updated in the Status Field of the RXRAM.

Sequencer clears the RXEMPTY bit while writing a new Received Data in the RXRAM. If the RXEMPTY bit is already 0, then the Sequencer sets the RCVR_OVRN bit to 1 to indicate that this particular location has been overwritten in the RXRAM.

23.2.3 DMA Requests

In order to reduce CPU overhead in handling SPI message traffic on a character-by-character basis, SPI can use the DMA controller to transfer the data

23.2.3.1 SPI/MibSPI Compatibility Mode DMA Requests

. The DMA request enable bit (DMA REQ EN) controls the assertion of requests to the DMA controller module. When a character is being transmitted or received, the SPI will signal the DMA via the DMA request signals, TX_DMA_REQ and RX_DMA_REQ. The DMA controller will then perform the required data transfer.

For efficient behavior during DMA operations, the transmitter empty and receive-buffer full interrupts can be disabled. For specific DMA features, see the DMA controller specification.

The SPI generates a request on the TX_DMA_REQ line each time the TX data is copied to the TX shift register either from the TXBUF or from peripheral data bus (when TXBUF is empty).

The first TX_DMA_REQ pulse is generated when either of the following is true:

- DMA REQ EN (SPIINT0[16]) is set to 1 while SPIEN (SPIGCR1[24]) is already 1.
- SPIEN (SPIGCR1[24]) is set to 1 while DMA REQ EN (SPIINT0[16]) is already 1.

The SPI generates a request on the RX_DMA_REQ line each time the received data is copied to the SPIBUF.

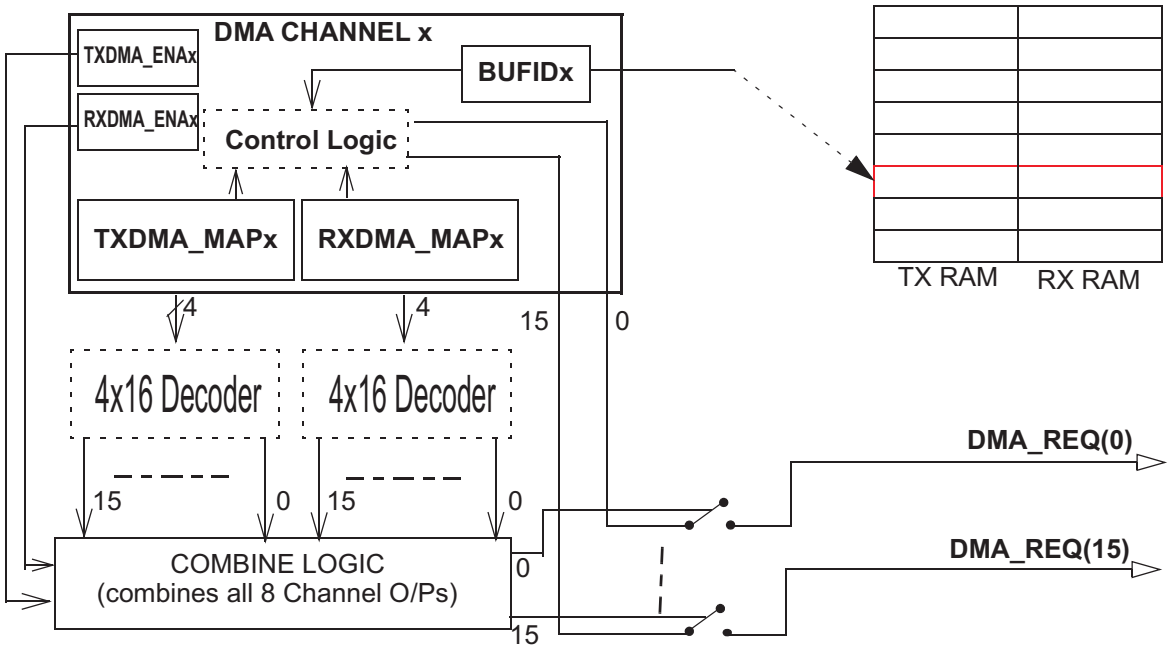
23.2.3.2 DMA in Multi-Buffer Mode

The MibSPI provides sophisticated programmable DMA control logic that completely eliminates the necessity of CPU intervention for data transfers, once programmed. When the multi-buffer mode is used, the DMA enable bit in the SPIINT0 register is ignored. DMA source or destination should be only the multi-buffer RAM and not SPIDAT0 / SPIDAT1 or SPIBUF register as in case of compatibility mode DMA.

The MibSPI offers up to eight DMA channels (for SEND and RECEIVE). All of the DMA channels are programmable individually and can be hooked to any buffer. The MibSPI provides up to 16 DMA request lines, and DMA requests from any channel can be programmed to be routed through any of these 16 lines. A DMA transfer can trigger both transmit and receive.

Each DMA channel has the capability to transfer a block of up to 32 data words without interruption using only one buffer of the array by configuring the DMAxCTRL register. Using the DMAxCOUNT and DMACTNTLEN register, up to 65535 (64K) words of data can be transferred without any interruption using just one buffer of the array. This enables the transfer of memory blocks from or into an external SPI memory.

Figure 23-3. DMA Channel and Request Line (Logical) Structure in Multi-buffer Mode



23.2.4 Interrupts

There are two levels of vectorized interrupts supported by the SPI. These interrupts can be caused under the following circumstances:

- Transmission error
- Receive overrun
- Receive complete (receive buffer full)
- Transmit buffer empty

These interrupts may be enabled or disabled via the SPIINT0 register.

During transmission, if one of the following errors occurs: BITERR, DESYNC, DLENERR, PARITYERR, or TIMEOUT, the corresponding bit in the SPIFLG register is set. If the corresponding enable bit is set, then an interrupt is generated. The level of all the above interrupts is set by the bit fields in the SPILVL register.

The error interrupts are enabled and prioritized independently from each other, but the interrupt generated will be the same if multiple errors are enabled on the same level. The SPIFLG register should be used to determine the actual cause of an error.

NOTE: Since there are two interrupt lines, one each for Level 0 and Level 1, it is possible for a programmer to separate out the interrupts for receive buffer full and transmit buffer empty. By programming one to Level 0 and the other to Level 1, it is possible to avoid a check on whether an interrupt occurred for transmit or for receive.A programmer can also choose to group all of the error interrupts into one interrupt line and both TX-empty and RX-full interrupts into another interrupt line using the LVL control register. In this way, it is possible to separate error-checking from normal data handling.

23.2.4.1 Interrupts in Multi-Buffer Mode

In multi-buffer mode, the SPI can generate interrupts on two levels.

In normal multi-buffer operation, the receive and transmit are not used and therefore the enable bits of SPIINT0 are not used.

The interrupts available in multi-buffer mode are:

- Transmission error interrupt
- Receive overrun interrupt
- TG suspended interrupt
- TG completed interrupt

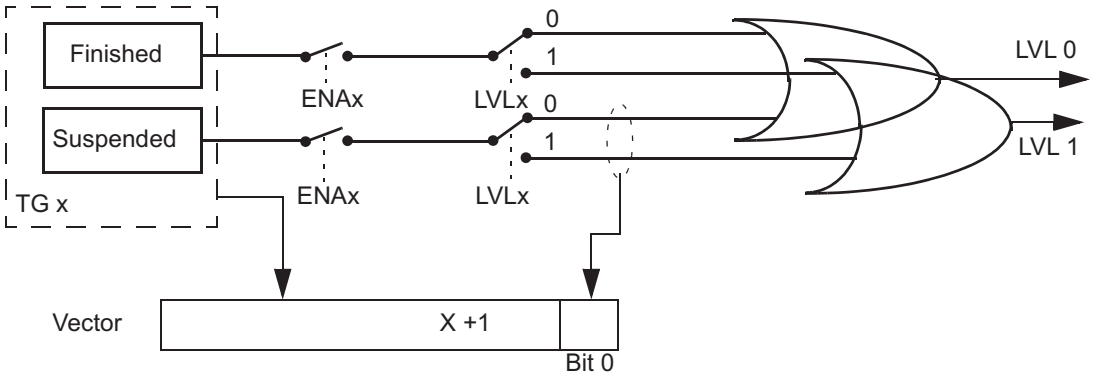
When a TG has finished and the corresponding enable bit in the TGINTENA register is set, a transfer-finished interrupt is generated. The level of priority of the interrupt is determined by the corresponding bit in the TGINTLVL register.

When a TG is suspended by a buffer that has been set as suspend to wait until TXFULL flag or/and RXEMPTY flag are set, and if the corresponding bit in the TGINTENA register is set, an transfer-suspended interrupt is generated. The level of priority of the interrupt is determined by the corresponding bit in the TGINTLVL register.

Figure 23-4 illustrates the TG interrupts.

During transmission, if one of the following errors occurs, BITERR, DESYNC, PARITYERR, TIMEOUT, DLENERR, the corresponding flag in the SPIFLG register is set. If the enable bit is set, then an interrupt is generated. The level of the interrupts could be generated according to the bit field in SPILVL register.

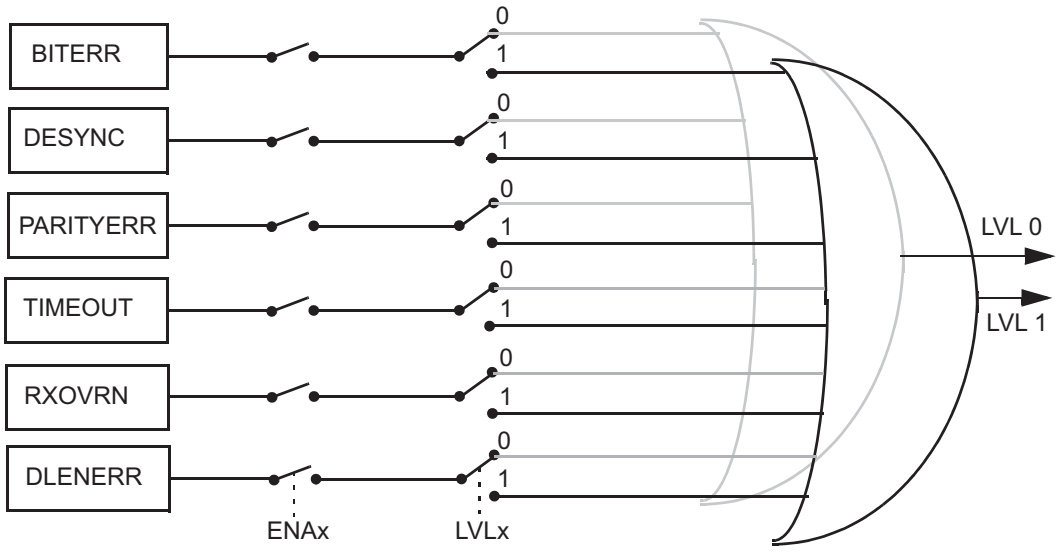
Figure 23-4. TG Interrupt Structure



The RXOVRN interrupt is generated when a buffer in the RXRAM is overwritten by a new received word. While writing newly received data to a RXRAM location, if the RXEMPTY bit of the corresponding location is 0, then the RXOVR bit will be set to 1 during the write operation, so that the buffer starts to indicate an overrun. This RXOVR flag is also reflected in SPIFLG register as RXOVRNINTFLG and the corresponding vector number is updated in TGINTVECT0/TGINTVECT1 register. If an overrun interrupt is enabled, then an interrupt will be generated indicating an overrun condition.

The error interrupts are enabled and prioritized independently from each other, but the vector generated by the SPI will be the same if multiple errors are enabled on the same level.

Figure 23-5. SPIFLG Interrupt Structure



Since the priority of an error interrupt is lower than a completion/suspend interrupt for a TG, the interrupts can be split into two levels. By programming all the error interrupts into Level 0 and TG-complete / TG-suspend interrupts into Level 1, it is possible to get a clear indication of the source of error interrupts. However, when a vector register shows an error interrupt, the actual buffer for which the error has occurred is not readily identifiable. Since each buffer in the multi-buffer RAM is stored along with its individual status flags, each buffer should be read until a buffer with any error flag set is found.

A separate interrupt line is provided to indicate the uncorrectable error condition in the MibSPI. This line is available (and valid) only in the multi-buffer mode of the MibSPI module and if the parity error detection feature for multi-buffer RAM is enabled.

23.2.5 Physical Interface

The SPI can be configured via software to operate as either a master or a slave. The MASTER bit (SPIGCR1[0]) selects the configuration of the SPISIMO and SPISOMI pins. CLKMOD bit (SPIGCR1[1]) determines whether an internal or external clock source will be used.

The slave chip select ($\overline{\text{SPISCS}}$[7:0]) pins, are used when communicating with multiple slave devices. When the a write occurs to SPIDAT1 in master mode, the $\overline{\text{SPISCS}}$ pins are automatically driven to select the specified slave.

Handshaking mechanism, provided by the $\overline{\text{SPIENA}}$ pin, enables a slave SPI to delay the generation of the clock signal supplied by the master if it is not prepared for the next exch ange of data.

23.2.5.1 Three-Pin Mode

In master mode configuration (MASTER = 1 and CLKMOD = 1), the SPI provides the serial clock on the SPICLK pin. Data is transmitted on the SPISIMO pin and received on the SPISOMI pin (see Figure 23-6).

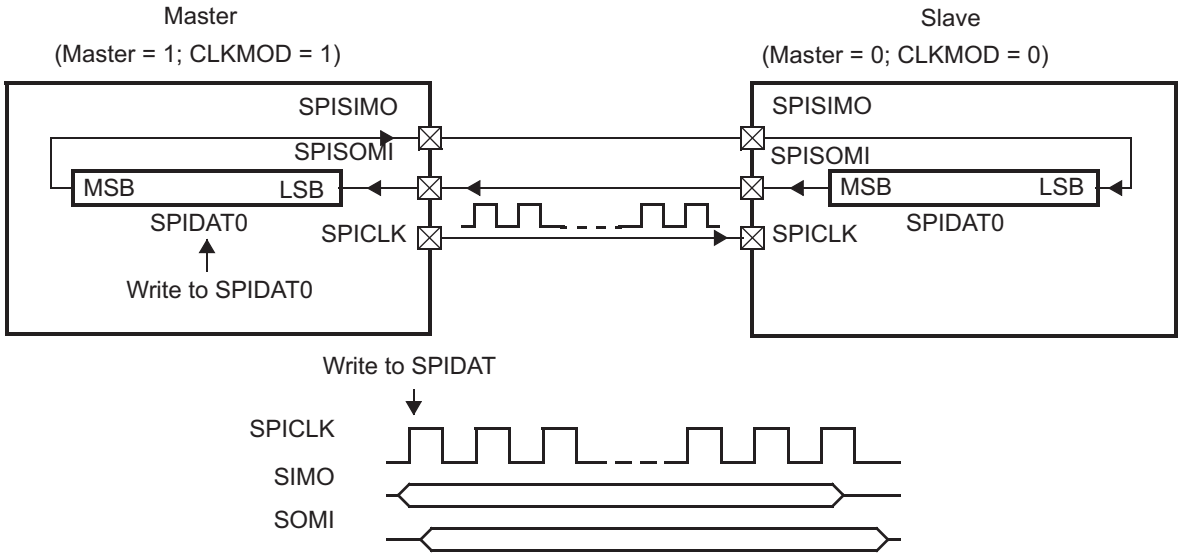
Data written to the shift register (SPIDAT0 / SPIDAT1) initiates data transmission on the SPISIMO pin, MSB first. Simultaneously, received data is shifted through the SPISOMI pin into the LSB of the SPIDAT0 register. When the selected number of bits have been transmitted, the received data in the shift register is transferred to the SPIBUF register for the CPU to read. Data is stored right-justified in SPIBUF.

See Section 23.2.1.2.2 and Section 23.2.2 for details about the data handling for transmit and receive operations.

In slave mode configuration (MASTER = 0 and CLKMOD = 0), data shifts out on the SPISOMI pin and in on the SPISIMO pin. The SPICLK pin is used as the input for the serial shift clock, which is supplied from the external network master. The transfer rate is defined by this clock.

Data written to the SPIDAT0 or SPIDAT1 register is transmitted to the network when the SPICLK signal is received from the network master. To receive data, the SPI waits for the network master to send the SPICLK signal and then shifts data on the SPISIMO pin into the RX shift register. If data is to be transmitted by the slave simultaneously, it must be written to the SPIDAT0 or SPIDAT1register before the beginning of the SPICLK signal.

Figure 23-6. SPI Three-Pin Operation



23.2.5.2 Four-Pin Mode with Chip Select

The three-pin option and the four-pin options of the SPI / MibSPI are identical in the master mode (CLKMOD = 1), except that the four-pin option uses either $\overline{\text{SPIENA}}$ or $\overline{\text{SPISCS}}$ [7:0] pins. The I/O directions of these pins are determined by the CLKMOD control bit as SPI / MibSPI and not general purpose I/O.

23.2.5.2.1 Four-Pin Option with $\overline{\text{SPISCS}}$

In master mode, each chip select signal is used to select a specific slave. In slave mode, chip select signal is used to enable/disable the transfer. Chip-select functionality is enabled by setting one of the $\overline{\text{SPISCS}}$ [7:0] pins as chip selects. It is disabled by setting all $\overline{\text{SPISCS}}$ [7:0] pins as GIOs in SPIPC0[3:0]

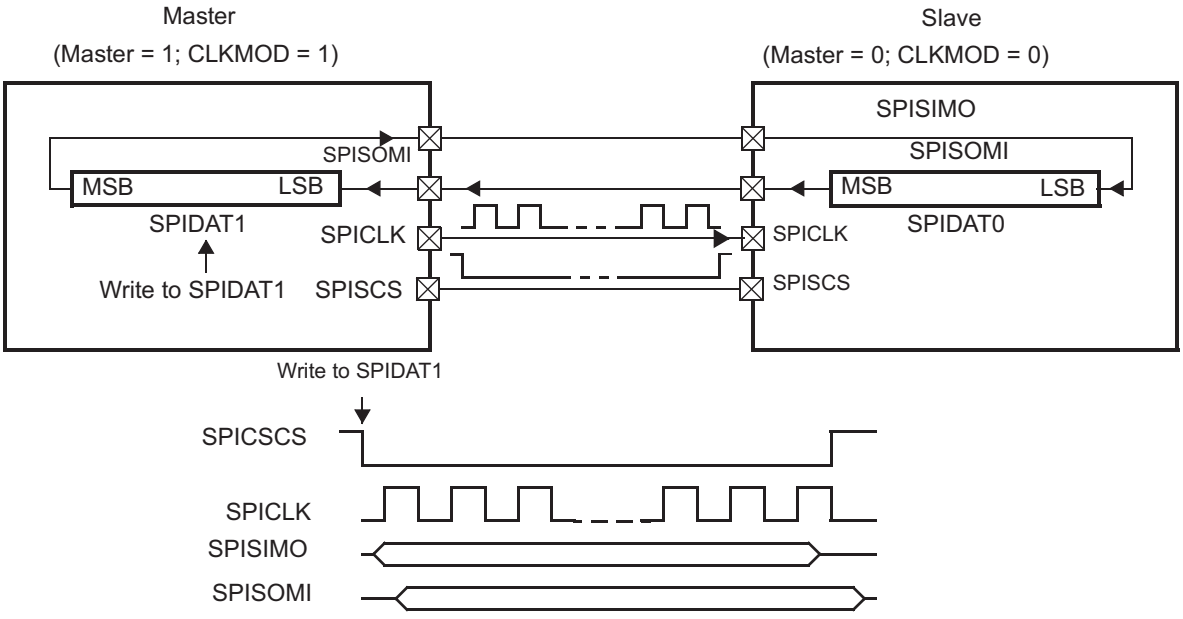
23.2.5.2.1.1 Multiple Chip Selects

The $\overline{\text{SPISCS}}$ [7:0] pins that are used must be configured as functional pins in the SPIPC0[7:0] register. The default pattern to be put on the $\overline{\text{SPISCS}}$ [7:0] when all the slaves are deactivated is set in the SPIDEF register. This pattern allows different slaves with different chip-select polarity to be activated by the SP/MibSPI.

The master-mode SPI is capable of driving either 0 or 1 as the active value for any $\overline{\text{SPISCS}}$[3:0] output pin. The drive state for $\overline{\text{SPISCS}}$[7:0] pins is controlled by the CSNR field of SPIDAT1. The pattern that is driven will select the slave to which the transmission is dedicated.

In slave mode, the SPI can only be selected by an active value of 0 on any of its selected $\overline{\text{SPISCS}}$ input pin.

Figure 23-7. Operation with $\overline{\text{SPISCS}}$



23.2.5.2.2 Four-Pin Option with $\overline{\text{SPIENA}}$

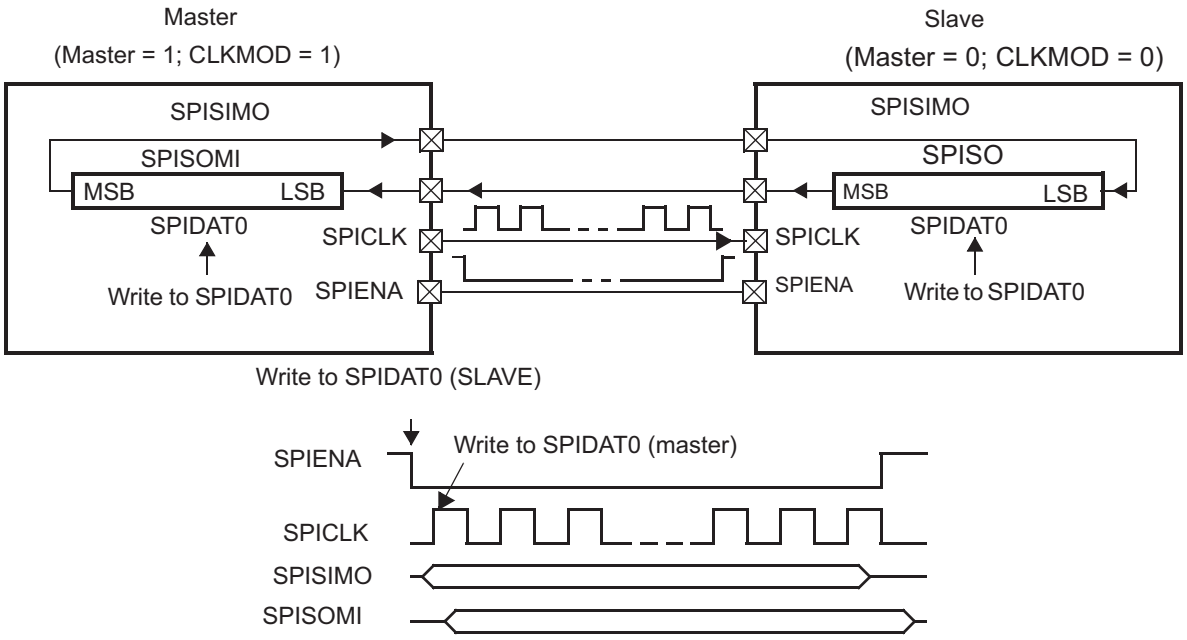
The $\overline{\text{SPIENA}}$ operates as a WAIT signal pin. For both the slave and the master, the $\overline{\text{SPIENA}}$ pin must be configured to be functional (SPIPC0[8] = 1). In this mode, an active low signal from the slave on the $\overline{\text{SPIENA}}$ pin allows the master SPI to drive the clock pulse stream. A high signal tells the master to hold the clock signal (and delay SPI activity).

If the $\overline{\text{SPIENA}}$ pin is in high-z mode (ENABLE_HIGHZ = 1), the slave will put $\overline{\text{SPIENA}}$ into the high-impedance once it completes receiving a new character. If the $\overline{\text{SPIENA}}$ pin is in push-pull mode (ENABLE_HIGHZ = 0), the slave will drive $\overline{\text{SPIENA}}$ to 1 once it completes receiving a new character. The slave will drive $\overline{\text{SPIENA}}$ low again for the next word to transfer, after new data is written to the slave TX shift register.

In master mode (CLKMOD = 1), if the $\overline{\text{SPIENA}}$ pin is configured as functional, then the pin will act as an input pin. If configured as a slave SPI and as functional, the $\overline{\text{SPIENA}}$ pin acts as an output pin.

NOTE: During a transfer, if a slave-mode SPI detects a deassertion of its chip select before its internal character length counter overflows, then it places SPISOMI and $\overline{\text{SPIENA}}$ (if ENABLE_HIGHZ bit is set to 1) in high-z mode. Once this condition has occurred, if a SPICLK edge is detected while the chip select is deasserted, then the SPI stops that transfer and sets an DLENERR error flag and generates an interrupt (if enabled).

Figure 23-8. Operation with $\overline{\text{SPIENA}}$



23.2.5.3 Five-Pin Operation (Hardware Handshaking)

Five-pin operation combines the functionality of three-pin mode, plus the enable pin and one or more chip select pins. The result is full hardware handshaking. To use this mode, both the SPIENA pin and the required number of SPISCS [3:0] pins must be configured as functional pins.

If the SPIENA pin is in high-z mode (ENABLE_HIGHZ = 1), the slave SPI will put this signal into the high-impedance state by default. The slave will drive the signal SPIENA low when new data is written to the slave shift register and the slave has been selected by the master (SPISCS is low).

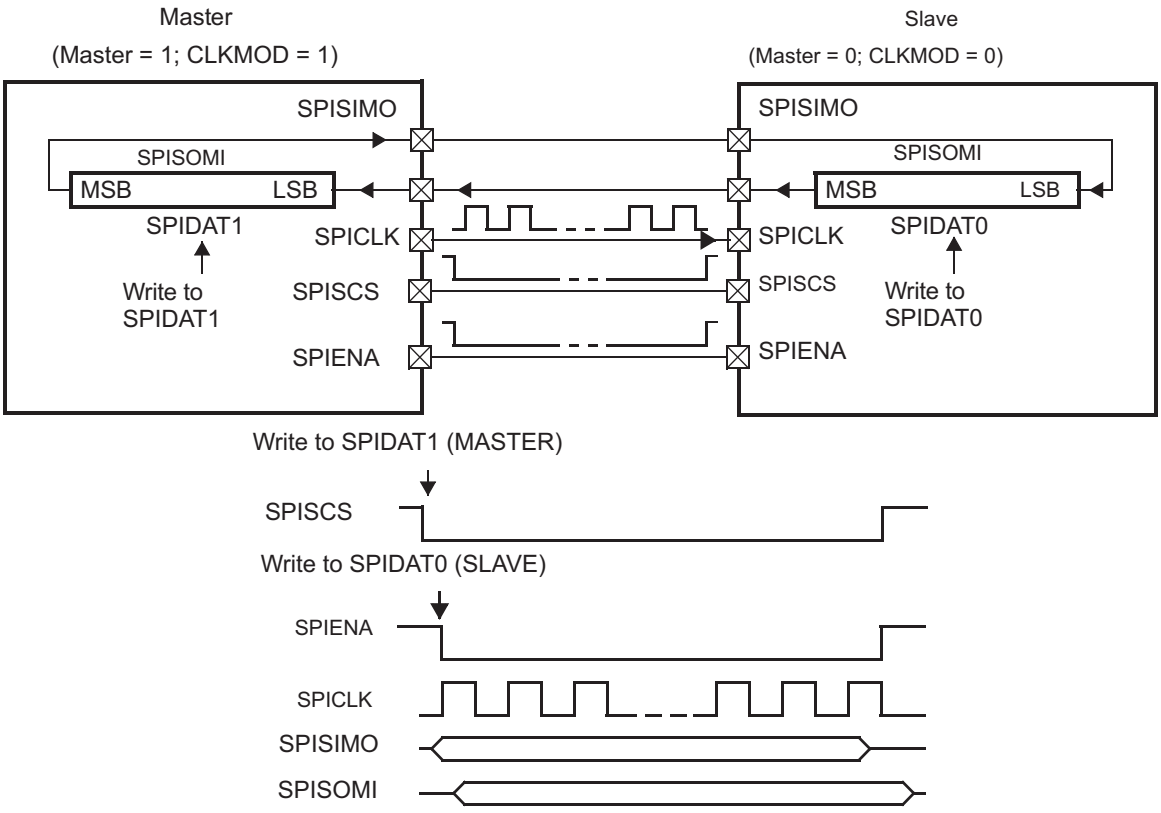
If the SPIENA pin is in push-pull mode (ENABLE_HIGHZ = 0), the slave SPI drives this pin high by default when it is in functional mode. The slave SPI will drive the SPIENA signal low when new data is written to the slave shift register (SPIDAT0/SPIDAT1) and the slave is selected by the master (SPISCS is low). If the slave is deselected by the master (SPISCS goes high), the slave SPIENA signal is driven high.

NOTE: Push-pull mode of the SPIENA pin can be used only when there is a single slave in the system. When multiple SPI slave devices are connected to the common SPIENA pin, all of the slaves should configure their SPIENA pins in high-Z mode.

In master mode, if the SPISCS pins are configured as functional pins, then the pins will be in output mode. A write to the master's SPIDAT1/SPIDAT0 register will automatically drive the SPISCS signals low. The master will drive the SPISCS signals high again after completing the transfer of the bits of the data.

In slave mode (CLKMOD = 0), the SPISCS pins will act as SPI functional inputs.

Figure 23-9. SPI Five-Pin Option with SPIENA and SPISCS



23.2.6 Advanced Module Configuration Options

23.2.6.1 Data Formats

To support multiple different types of slaves in one SPI network, four independent data word formats are implemented that allow configuration of individual data word length, polarity, phase, and bit rate. Each word transmitted can select which data format to use via the bits DFSEL[1:0] in its control field from one of the four data word formats. Same data format can be supported on multiple chip selects.

Data formats 0, 1, 2, and 3 can be configured through SPIFMTx control registers.

Each SPI data format includes the standard SPI data format with enhanced features:

- Individually-configurable shift direction can be used to select MSB first or LSB first, whereas the position of the MSB depends on the configured data word length.
- Receive data is automatically right-aligned, independent of shift direction and data word length. Transmit data has to be written right-aligned into the SPI and the internal shift register will transmit according to the selected shift direction and data word length for correct transfer.
- To increase fault detection of data transmission and reception, an odd or even parity bit can be added at the end of a data word. The parity generator can be enabled or disabled individually for each data format. If a received parity bit does not match with the locally calculated parity bit, the parity error flag (PARITYERR) is set and an interrupt is asserted (if enabled).

Since the master-mode SPI can drive two consecutive accesses to the same slave, an 8-bit delay counter is available to satisfy the delay time for data to be refreshed in the accessed slave. The delay counter can be programmed as part of the data format.

CHARLEN[4:0] specifies the number of bits (2 to 16) in the data word. The CHARLEN[4:0] value directs the state control logic to count the number of bits received or transmitted to determine when a complete word is transferred.

Data word length **must** be programmed to the same length for both the **master** and the **slave**. However, when chip selects are used, there may be multiple targets with different lengths in the system.

NOTE: Data must be right-justified when it is written to the SPI for transmission irrespective of its character length or word length.

Figure 23-10 shows how a 12-bit word (0xEC9) needs to be written to the transmit buffer to be transmitted correctly.

Figure 23-10. Format for Transmitting an 12-Bit Word

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	1	1	1	0	1	1	0	0	1	0	0	1

NOTE: The received data is always stored right-justified regardless of the character length or direction of shifting and is padded with leading 0s when the character length is less than 16 bits.

Figure 23-11 shows how a 10-bit word (0x0A2) is stored in the buffer once it is received.

Figure 23-11. Format for Receiving an 10-Bit Word

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0

23.2.6.2 Clocking Modes

SPICLK may operate in four different modes, depending on the choice of phase (delay/no delay) and the polarity (rising edge/falling edge) of the clock.

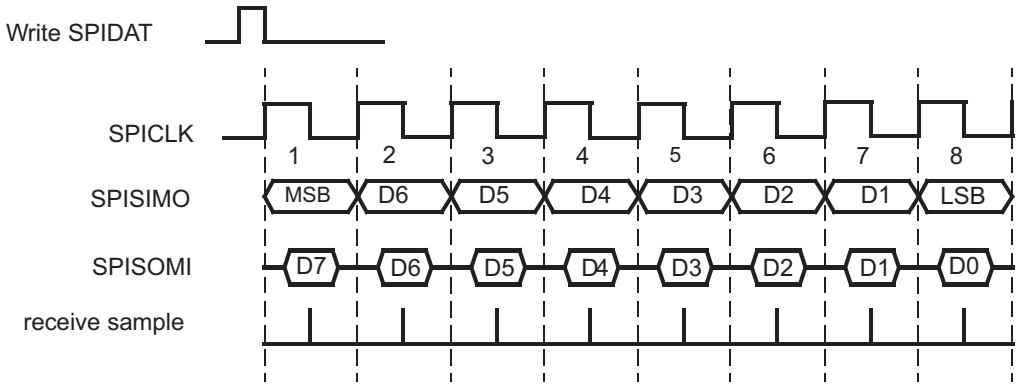
The data input and output edges depend on the values of both POLARITY and PHASE as shown in Table 23-5.

Table 23-5. Clocking Modes

POLARITY	PHASE	Action
0	0	Data is output on the rising edge of SPICLK. Input data is latched on the falling edge.
0	1	Data is output one half-cycle before the first rising edge of SPICLK and on subsequent falling edges. Input data is latched on the rising edge of SPICLK.
1	0	Data is output on the falling edge of SPICLK. Input data is latched on the rising edge.
1	1	Data is output one half-cycle before the first falling edge of SPICLK and on subsequent rising edges. Input data is latched on the falling edge of SPICLK.

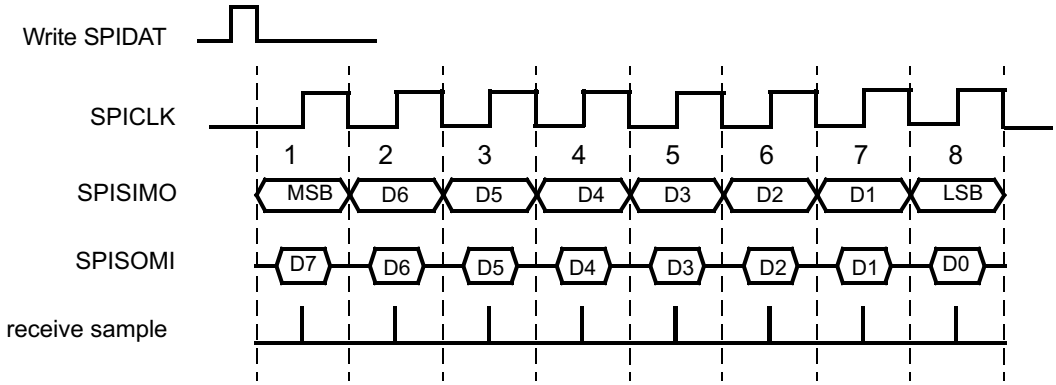
Figure 23-12 to Figure 23-15 illustrate the four possible configurations of **SPICLK** corresponding to each mode. Having four signal options allows the SPI to interface with many different types of serial devices.

Figure 23-12. Clock Mode with Polarity = 0 and Phase = 0



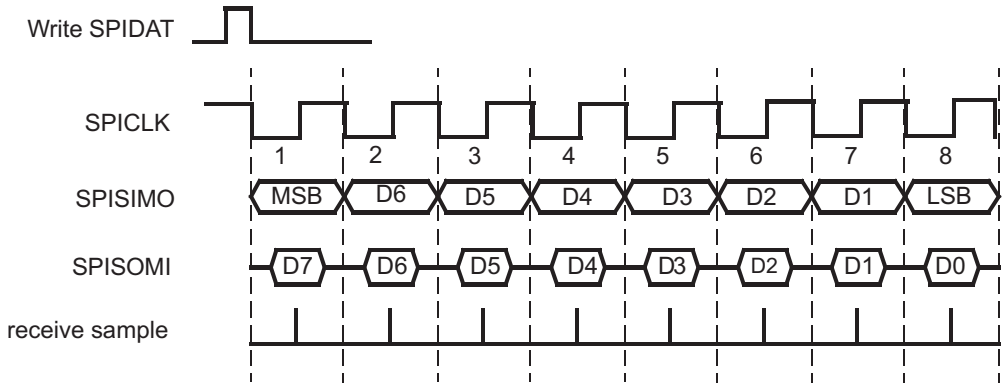
Data is output on the rising edge of SPICLK.
Input data is latched on the falling edge of SPICLK.

Figure 23-13. Clock Mode with Polarity = 0 and Phase = 1



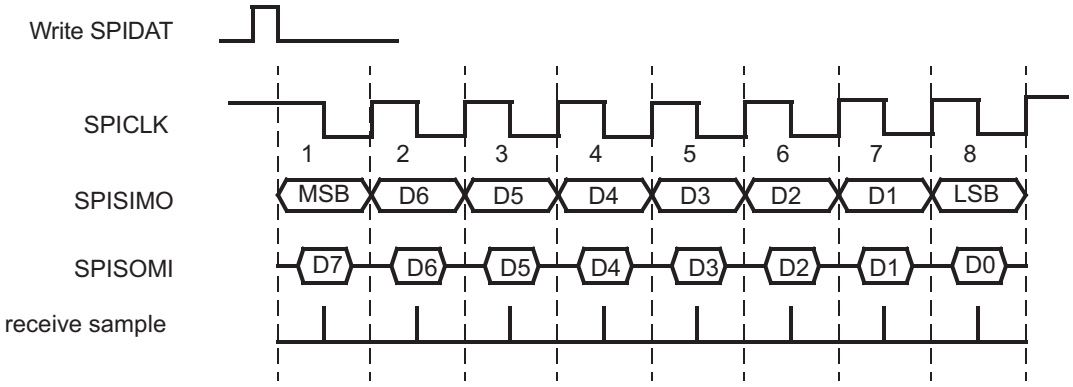
Data is output one-half cycle before the first rising edge of SPICLK and on subsequent falling edges of SPICLK
Input data is latched on the rising edge of SPICLK

Figure 23-14. Clock Mode with Polarity = 1 and Phase = 0



Data is output on the falling edge of SPICLK.
Input data is latched on the rising edge of SPICLK.

Figure 23-15. Clock Mode with Polarity = 1 and Phase = 1

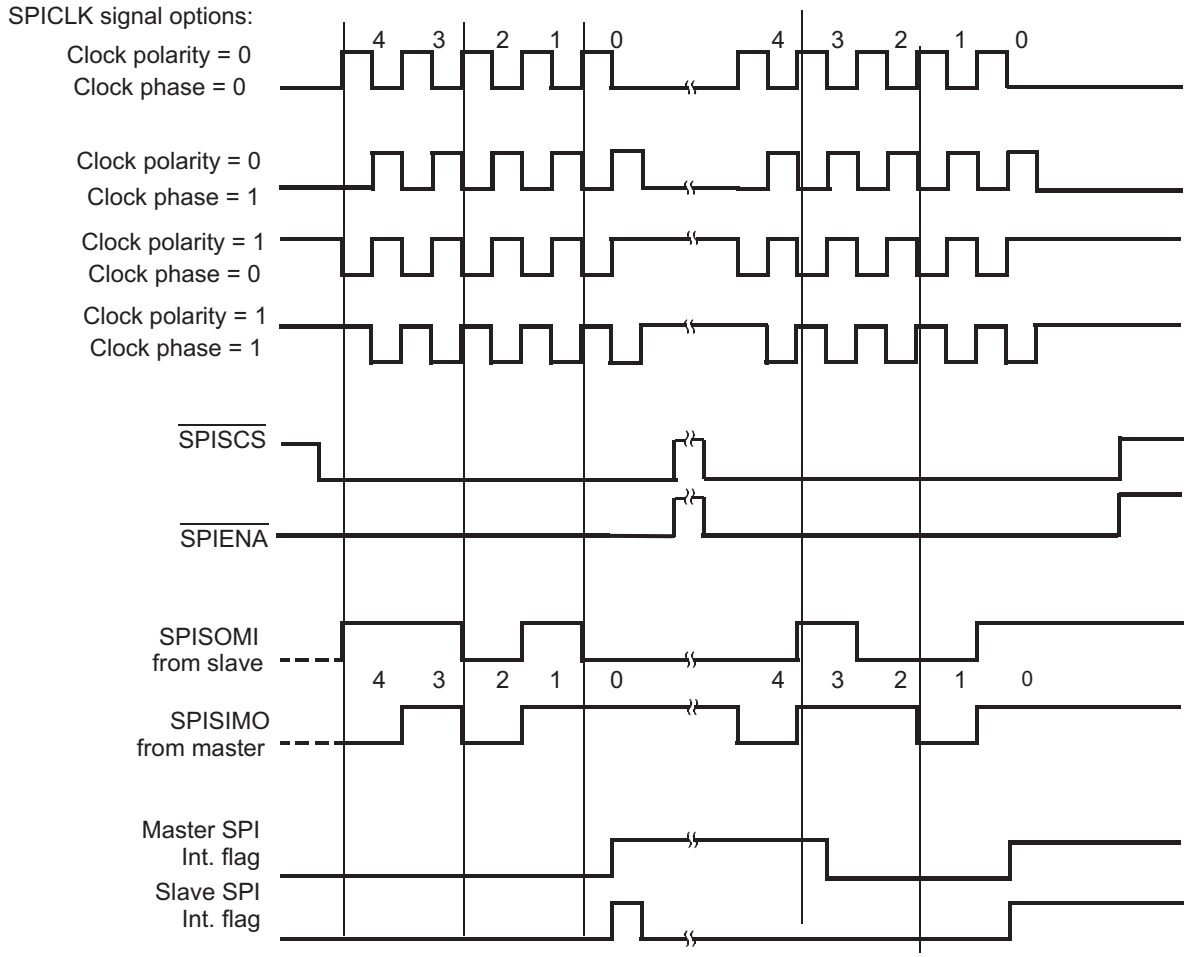


Data is output one-half cycle before the first falling edge of SPICLK and on the subsequent rising edges of SPICLK.
Input data is latched on the falling edge of SPICLK.

23.2.6.2.1 Data Transfer Example

Figure 23-16 illustrates a SPI data transfer between two devices using a character length of five bits.

Figure 23-16. Five Bits per Character (5-Pin Option)



23.2.6.3 Decoded and Encoded Chip Select (Master Only)

In this device the SPI can connect to up to 4 individual slave devices using chip-selects by routing one wire to each slave. The 4 chip selects in the control field are directly connected to the 4 pins. The default value of each chip select (not active) can be configured via the register CSDEF. During a transmission, the value of the chip select control field (CSNR[7:0]) of the SPIDAT1 register (SPIDAT1[23:16]) is driven on the $\overline{\text{SPISCS}}$ [4:0] pins. When the transmission finishes the default chip-select value (defined by the CSDEF register) is put on the $\overline{\text{SPISCS}}$ [4:0] pins.

The SPI can support more than 4 slaves by using encoded chip selects. To connect the SPI with encoded slaves devices, the CSNR field allows multiple active $\overline{\text{SPISCS}}$ pins at the same time, which enables binary encoded chip selects from 0 to 16. To use encoded chip selects, all four chip select lines have to be connected to each slave device and each slave needs to have a unique chip-select address. The CSDEF register is used to provide the address at which slaves devices are all de-selected.

Users can combine decoded and encoded chip selects. For example, *n* $\overline{\text{SPISCS}}$ pins can be used for encoding a *n*-bit address and the remaining pins can be connected to decoded-mode slaves.

23.2.6.4 Chip Select Timing Control

This section describes few fields of the contrlol register SPIDELAY this register decides the chip select and timing control for the device

23.2.6.4.1 Chip-Select-Active-to-Transmit-Start-Delay (C2TDELAY)

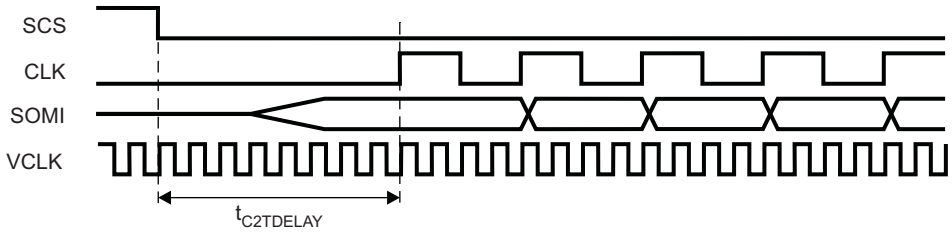
C2TDELAY is used in master mode only. It defines a setup time for the slave device that delays the data transmission from the chip select active edge by a multiple of VBUSPCLK cycles. ChipSelect-active-to-transmission delays between 2 to 257 VBUSPCLK cycles can be achieved.

The setup time value is calculated as:

$t_{C2TDELAY}$= (C2TDELAY + 2) × VCLK Period

Figure 23-17 is the timing diagram when C2TDELAY of 8 VCLK Cycles.

Figure 23-17. Example: $t_{C2TDELAY}$= 8 VCLK Cycles



23.2.6.4.2 Transmit-End-to-Chip-Select-Inactive-Delay (T2CDELAY)

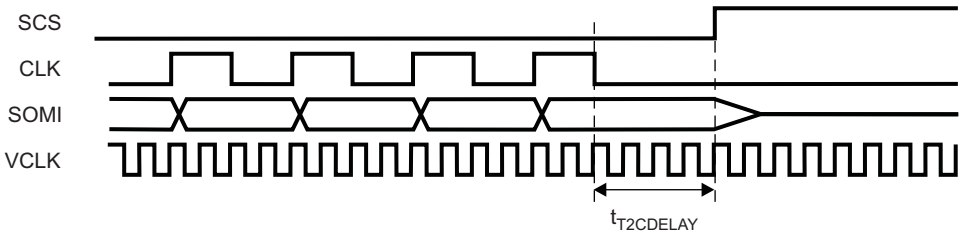
T2CDELAY is used in master mode only. It defines a hold time for the slave device that delays the chip select deactivation by a multiple of VBUSPCLK cycles after the last bit is transferred. T2CDELAY can be configured between 2 and 256 VBUSPCLK cycles.

The hold time value is calculated as:

$t_{T2CDELAY}$= (T2CDELAY +1) × VCLK Period

Figure 23-18 is the timing diagram when T2CDELAY of 4 VCLK Cycles.

Figure 23-18. Example: $t_{T2CDELAY}$= 4 VCLK Cycles



23.2.6.4.3 Transmit-Data-Finished-to-ENA-Pin-Inactive-Time-Out (T2EDELAY)

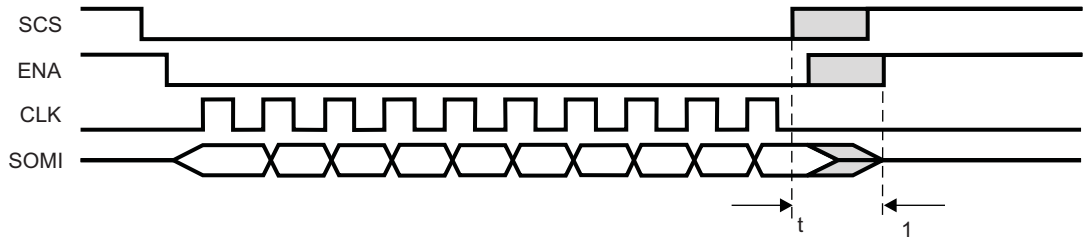
T2EDELAY is used in master mode only. It defines a time-out value as a multiple of SPI clock before the ENAble signal has to become inactive and after the CS becomes inactive. The SPI clock depends on which data format is selected. If the slave device is missing one or more clock edges, it is becoming de-synchronized. Although the master has finished the data transfer the

The T2EDELAY defines a time-out value that triggers the DESYNC flag, if the ENA signal isn't deactivated in time. DESYNC flag is set to indicate that the Slave device did not deassert its SPIENA pin in time to acknowledge that it has received all the bits of the sent character.

The timeout value is calculated as:

$t_{T2EDELAY}$ = T2EDELAY/SPIclock

Figure 23-19. Transmit-Data-Finished-to-ENA-Inactive-Timeout



NOTE: If T2CDELAY is programmed a non-zero value, then T2EDELAY will start only after the T2CDELAY completes. This should be taken into consideration to determine an optimum value of T2EDELAY.

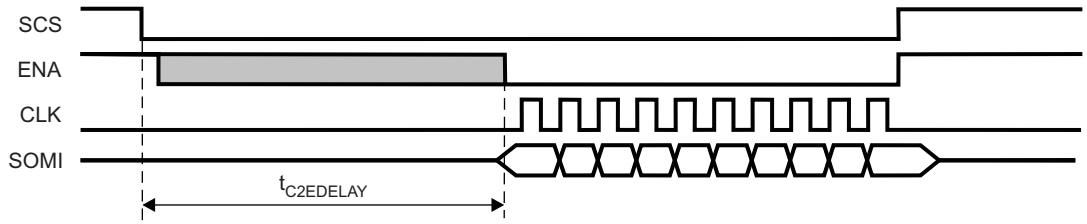
23.2.6.4.4 Chip-Select-Active-to-ENA-Signal-Active-Time-Out (C2EDELAY)

C2EDELAY is utilized only in master mode and it applies only if the addressed slave generates an ENA signal as a hardware handshake response. C2EDELAY defines the maximum time between the SPI / MibSPI activating the chip select signal and the addressed slave responding by activating the ENA signal. C2EDELAY defines a time-out value as a multiple of SPI clocks. The SPI clock depends on whether data format 0 or data format 1 is selected.

The timeout value is calculated as:

$t_{C2EDELAY}$ = C2EDELAY/SPIclock

Figure 23-20. Chip-Select-Active-to-ENA-Signal-Active-Timeout



NOTE:

- If the slave device is not responding with the ENA signal before the time-out value is reached, the TIMEOUT flag in SPIFLG register is set and an interrupt is asserted if enabled.
- If a time-out occurs the MibSPI clears the transmit request of the timed-out buffer, sets the TIMEOUT flag for the current buffer and continues with the transfer of the next buffer in the sequence that is enabled.
- If C2TDELAY is programmed a non-zero value, then C2EDELAY will start only after the C2TDELAY completes. This should be taken into consideration to determine an optimum value of C2EDELAY.

23.2.6.5 Multiple Transfers to Same Slave and Variable Chip Select Setup and Hold Timing

This section gives information on the Variable chip select setup and it shows how CSHOLD bit is used and how the multiple transfers to same slave is enabled in the devic

23.2.6.5.1 Variable Chip Select Setup and Hold Timing (Master Only)

In order to support slow slave devices a delay counter can be configured to delay data transmission after the chip select is activated. A second delay counter can be configured to delay the chip select deactivation after the last data bit is transferred. Both delay counters are clocked with peripheral clock(VCLK).

If a particular data format specifically does not require these additional set-up or hold times for the chip select pin(s), then they can be disabled in the corresponding SPIFMTx register.

23.2.6.5.2 Hold Chip-Select Active

Some slave devices require the chip select signal to be held continuously active during several consecutive data word transfers. Other slave devices require the chip select signal to be deactivated between consecutive data word transfers.

CSHOLD is programmable in both master and slave modes of the multi-buffer mode of SPI. However, the meaning of CSHOLD in master mode and slave mode are different.

NOTE: If the CSHOLD bit is set within the current data control field, the programmed hold time and the following programmed set-up time will not be applied between transactions.

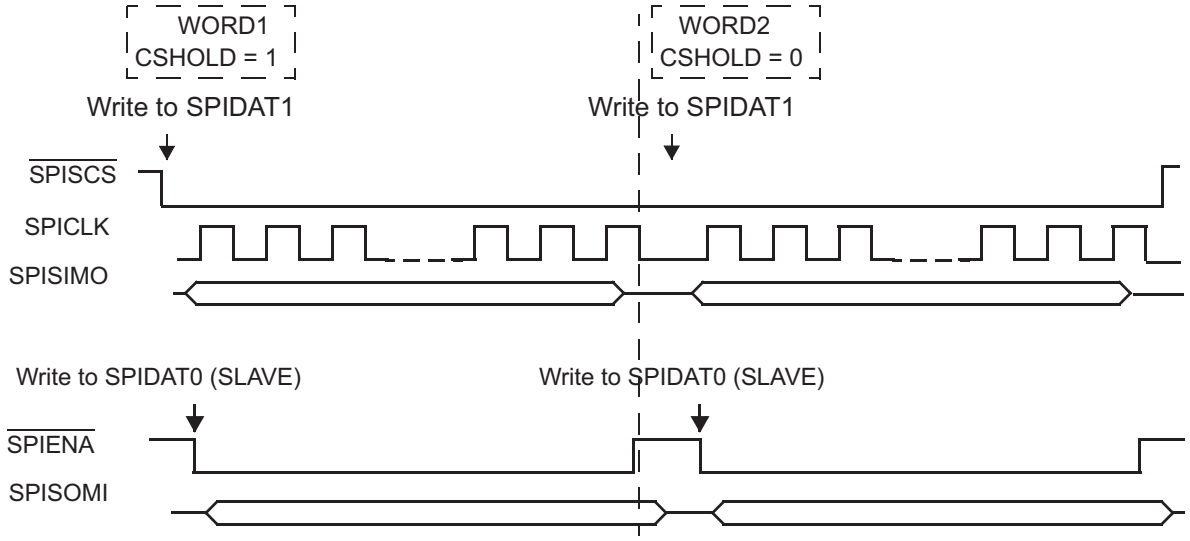
23.2.6.5.2.1 CSHOLD Bit in Master Mode

Each word in a master-mode SPI can be individually initialized for one of the two modes via the CSHOLD bit in its control field.

If the CSHOLD bit is set in the control field of a word, the chip select signal will not be deactivated until the next control field is loaded with new chip select information. Since the chip-select is maintained active between two transfers, the chip-select hold delay (T2CDELAY) is not applied at the end of the current transaction, and the chip-select set-up time delay (C2TDELAY) is not applied as well at the beginning of the following transaction. However, the wait delay (WDELAY) will be still applied between the two transactions, if the WDEL bit is set within the control field.

Figure 23-21 shows the SPI pins when a master-mode SPI transfers a word that has its CSHOLD bit set. The chip-select pins will not be deasserted after the completion of this word. If the next word to transmit has the same chip-select number (CSNR) value, the chip select pins will be maintained until the completion of the second word, regardless of whether the CSHOLD bit is set or not.

Figure 23-21. Typical Diagram when a Buffer in Master is in CSHOLD Mode (SPI-SPI)



23.2.6.5.2.2 CSHOLD Bit in Slave Mode (Multi-buffered Mode)

If the CSHOLD bit in a buffer is set to 1, then the MibSPI does not wait for the $\overline{\text{SPISCS}}$ pins to be de-activated at the end of the shift operation to copy the received data to the receive RAM. With this feature, it is possible for a slave in multi-buffer mode to do multiple data transfers without requiring the $\overline{\text{SPISCS}}$ pins to be deasserted between two buffer transfers.

If the CSHOLD bit in a buffer is cleared to 0 in a slave MibSPI, even after the shift operation is done, the MibSPI waits until the $\overline{\text{SPISCS}}$ pin (if functional) is deasserted to copy the received data to the RXRAM.

If the CSHOLD bit is maintained as 0 across all the buffers, then the slave in multi-buffer mode requires its $\overline{\text{SPISCS}}$ pins to be deasserted between any two buffer transfers; otherwise, the Slave SPI will be unable to respond to the next data transfer.

NOTE: In compatibility mode, the slave does not require the $\overline{\text{SPISCS}}$ pin to be deasserted between two buffer transfers. The CSHOLD bit of the slave will be ignored in compatibility mode.

23.2.6.6 Parallel Mode (Multiple SIMO/SOMI Support, not available on all devices)

In order to increase throughput, the parallel mode of the SPI enables the module to send data over more than one data line (Parallel 2, 4 or 8). When parallel mode is used, the data length must be set as 16 bits. Only module MIBSPIP5 supports Parallel Mode.

This feature increases throughput by 2 for 2 pins, by 4 for 4 pins, or by 8 for 8 pins.

Parallel mode supports the following features:

- Scalable data lines (1, 2, 4, 8) per direction. (SOMI and SIMO lines)
- All clock schemes are supported (clock phase and polarity)
- Parity is supported. The parity bit will be transmitted on bit0 of the SIMO/SOMI lines. The receive parity is expected on bit0 of the SOMI/SIMO pins.

Parallel mode can be programmed using the PMODEx[1:0] bits of SPIPMCTRL register. See for details about this register.

After reset the parallel mode selection bits are cleared (single SIMO/SOMI lines).

23.2.6.6.1 Parallel Mode Block Diagram

Figure 23-22 and Figure 23-23 show the parallel connections to the SPI shift register.

Figure 23-22. Block Diagram Shift Register, MSB First

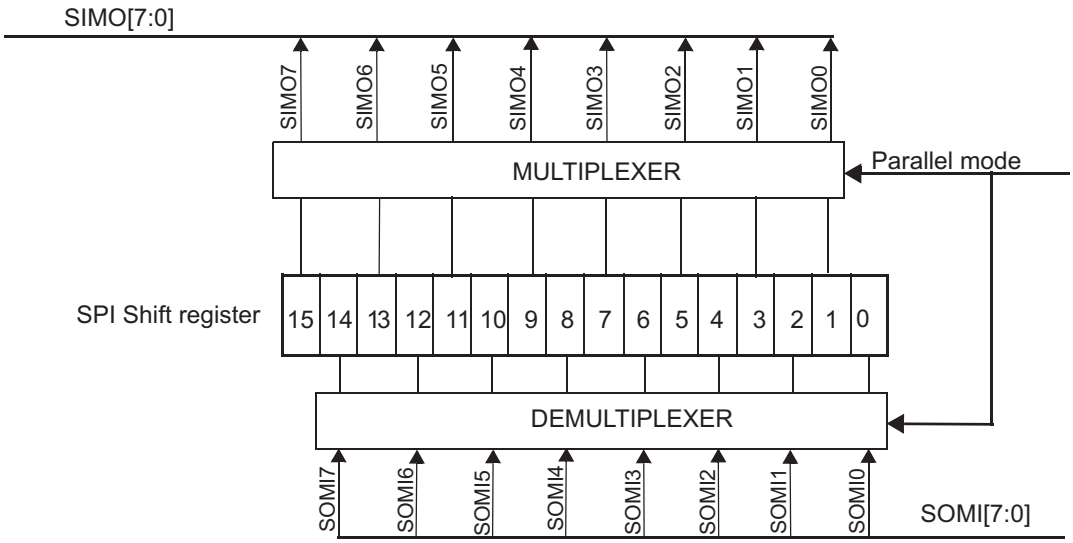
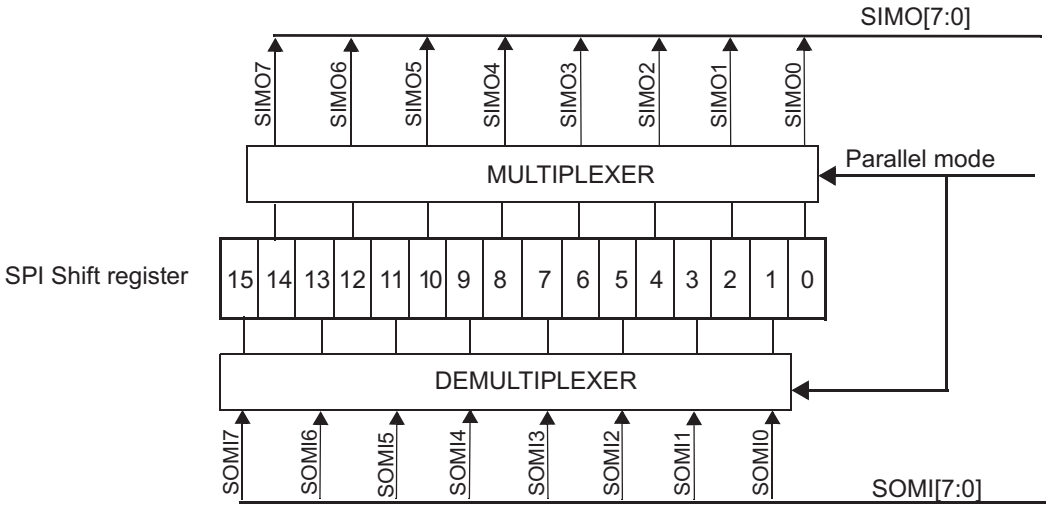


Figure 23-23. Block Diagram Shift Register, LSB First



23.2.6.6.2 Parallel Mode Pin Mapping, MSB First

Table 23-6 and Table 23-7 describe the SOMI and SIMO pin mapping when the SPI is used in parallel mode (1, 2, 4, 8) pin mode, MSB first.

NOTE: MSB-first or LSB-first can be configured using the SHIFTDIRx bit of the SPIFMTx registers.

Table 23-6. Pin Mapping for SIMO Pin with MSB First

Parallel Mode	Shift Register Bit	SIMO[7:0]
1	15	0
2	15	1
	7	0
4	15	3
	11	2
	7	1
	3	0
8	15	7
	13	6
	11	5
	9	4
	7	3
	5	2
	3	1
	1	0

Table 23-7. Pin Mapping for SOMI Pin with MSB First

Parallel Mode	Shift Register Bit	SOMI[7:0]
1	0	0
2	0	0
	8	1
4	0	0
	4	1
	8	2
	12	3
8	0	0
	2	1
	4	2
	6	3
	8	4
	10	5
	12	6
	14	7

23.2.6.6.3 Parallel Mode Pin Mapping, MSB-First, LSB-First

Table 23-8 and Table 23-9 describe the SIMO and SOMI pin mapping when SPI is used in parallel mode (1, 2, 4, 8) pin mode, LSB first.

Table 23-8. Pin Mapping for SIMO Pin with LSB First

Parallel Mode	Shift Register Bit	SIMO[7:0]
1	0	0
2	8	1
	0	0
4	12	3
	8	2
	4	1
	0	0
8	14	7
	12	6
	10	5
	8	4
	6	3
	4	2
	2	1
	0	0

Table 23-9. Pin Mapping for SOMI Pin with LSB First

Parallel Mode	Shift Register Bit	SOMI[7:0]
1	15	0
2	7	0
	15	1
4	3	0
	7	1
	11	2
	15	3
8	1	0
	3	1
	5	2
	7	3
	9	4
	11	5
	13	6
	15	7

23.2.6.6.4 2-Data Line Mode (MSB First, Phase 0, Polarity 0)

In 2-data line mode (master mode) the shift register bits 15 and 7 will be connected to the pins SIMO[1] and SIMO[0], and the shift register bits 8 and 0 will be connected to the pins SOMI[1] and SOMI[0] or vice versa in slave mode. After writing to the SPIDAT0/SPIDAT1 register, the bits 15 and 7 will be output on SIMO[1] and SIMO[0] on the rising edge if SPICLK. With the falling clock edge of the SPICLK, the received data on SOMI[1] and SOMI[0] will be latched to the shift register bits 8 and 0. The subsequent rising edge of SPICLK will shift the data in the shift register by 1 bit to the left. (SIMO[1] will shift the data out from bit 15 to 8, SIMO[0] will shift the data out from bit 7 to 0). After eight SPICLK cycles, when the full data word is transferred, the shift register (16 bits) is copied to the receive buffer, and the RXINT flag will be set. Figure 23-24 shows the clock /data diagram of the 2-data line mode. Figure 23-25 shows the timing of a two-pin parallel transfer.

Figure 23-24. 2-data Line Mode (Phase 0, Polarity 0)

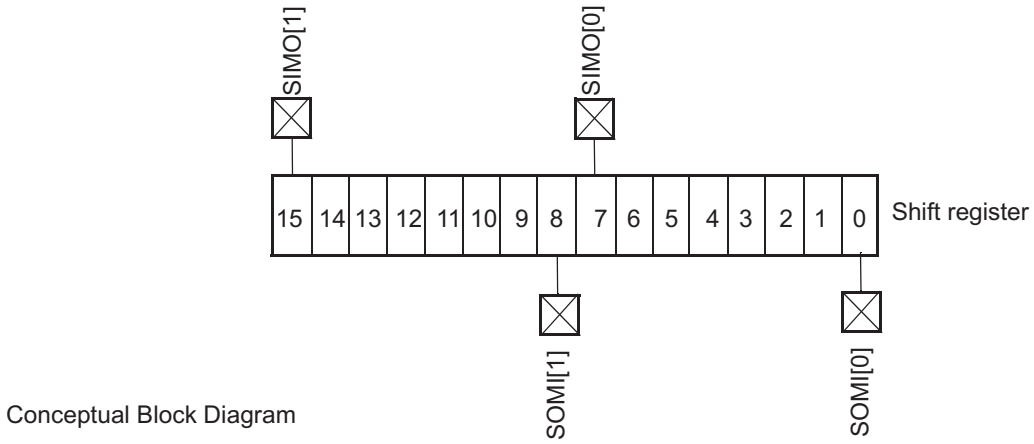
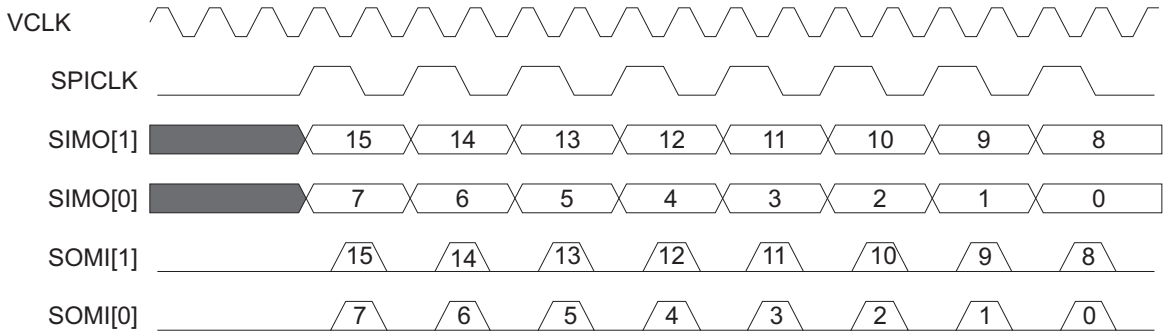


Figure 23-25. Two-Pin Parallel Mode Timing Diagram (Phase 0, Polarity 0)



23.2.6.6.5 4-Data Line Mode (MSB First, Phase 0, Polarity 0)

In 4-data line mode (master mode) the shift register bits 15, 11, 7, and 3 will be connected to the pins SIMO[3], SIMO[2], SIMO[1], and SIMO[0], and the shift register bits 12, 8, 4, and 0 will be connected to the pins SOMI[3], SOMI[2], SOMI[1], and SOMI[0] (or vice versa in slave mode). After writing to SPIDAT1/SPIDAT0, the bits 15, 11, 7, and 3 will be output on SIMO[3], SIMO[2], SIMO[1], and SIMO[0] on the rising edge of SPICLK. With the falling clock edge of the SPICLK, the received data on SOMI[3], SOMI[2], SOMI[1] and SOMI[0] will be latched to shift register bits 12, 8, 4, and 0. The subsequent rising edge of SPICLK will shift data in the shift register by 1 bit to the left (SIMO[3] will shift the data out from bit 15 to 12, SIMO[2] will shift the data out from bit 11 to 8, SIMO[1] will shift the data out from bit 7 to 4, SIMO[0] will shift the data out from bit 3 to 0). After four SPICLK cycles, when the full data word is transferred, the shift register (16 bits) is copied to the receive buffer, and the RXINT flag will be set.

Figure 23-26 shows the clock/data diagram of the four-data line mode. Figure 23-27, shows the timing diagram for four-data line mode.

Figure 23-26. 4-Data Line Mode (Phase 0, Polarity 0)

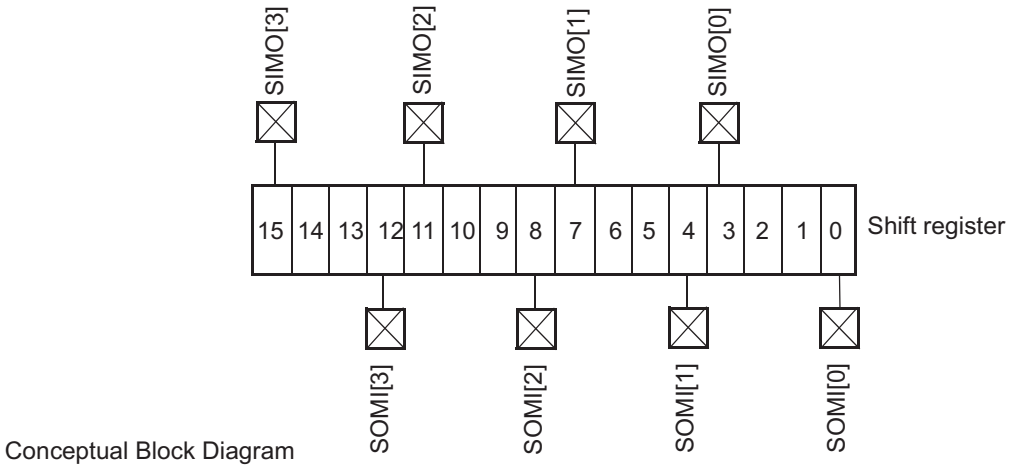
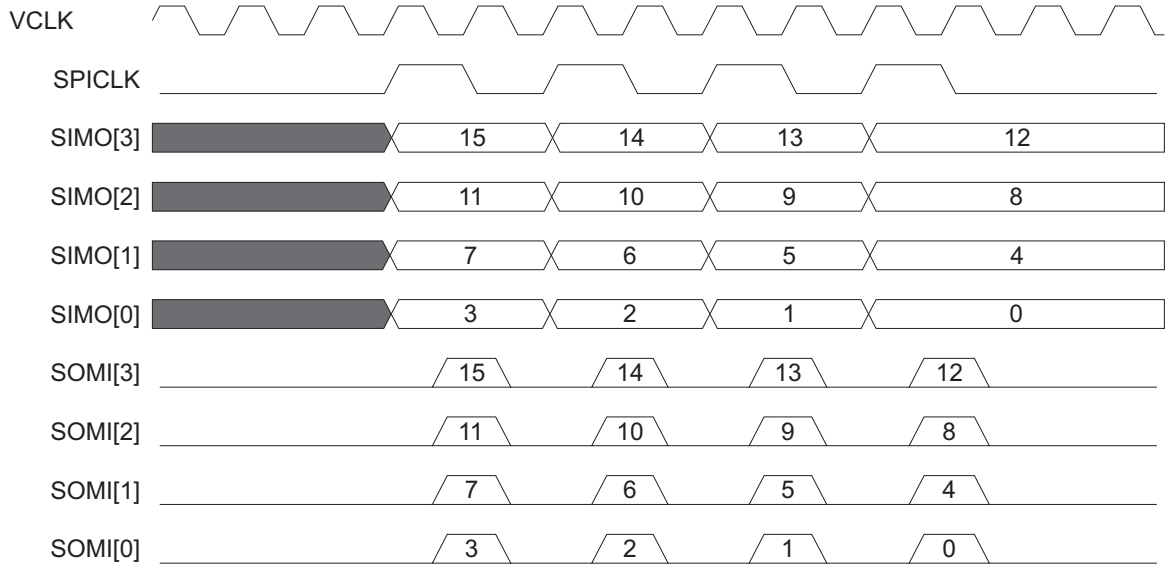


Figure 23-27. 4 Pins Parallel Mode Timing Diagram (Phase 0, Polarity 0)



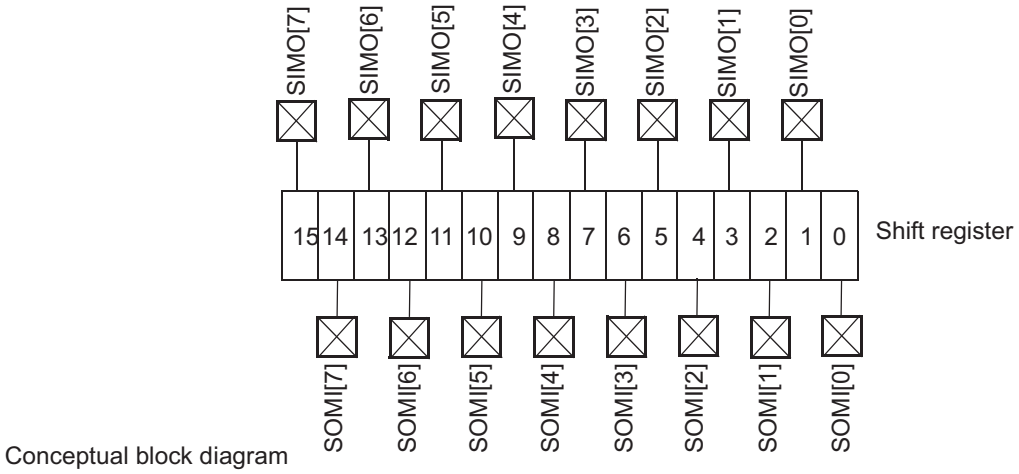
23.2.6.6.6 8-Data Line Mode (MSB First, Phase 0, Polarity 0)

In 8-data line mode (master mode) the shift register bits 15, 13, 11, 9, 7, 5 and 3 will be connected to the pins SIMO[7], SIMO[6], SIMO[5], SIMO[4], SIMO[3], SIMO[2], SIMO[1], and SIMO[0], and the shift-register bits 14, 12, 10, 8, 6, 4, and 0 will be connected to the pins SOMI[7], SOMI[6], SOMI[5], SOMI[4], SOMI[3], SOMI[2], SOMI[1], and SOMI[0] (or vice versa in slave mode).

After writing to SPIDAT0/SPIDAT1, the bits 15, 13, 11, 9, 7, 5, 3, and 1 will be output on SIMO[7], SIMO[6], SIMO[5], SIMO[4], SIMO[3], SIMO[2], SIMO[1], and SIMO[0], on the rising edge of SPICLK. On the falling clock edge of the SPICLK, the received data on SOMI[8], SOMI[7], SOMI[6],SOMI[5], SOMI[4], SOMI[3], SOMI[2], SOMI[1], and SOMI[0] will be latched to the shift register bits 14, 12, 10, 8, 6, 4, 2, and 0.

The subsequent rising edge of SPICLK will shift the data in the shift register by one bit to the left. After two SPICLK cycles, when the full data word is transferred the shift register (16 bits) is copied to the receive buffer, and the RXINT flag will be set. Figure 23-28 shows the clock/data diagram of the 8-data line mode. Figure 23-29 shows the pin timings for 8-data line mode.

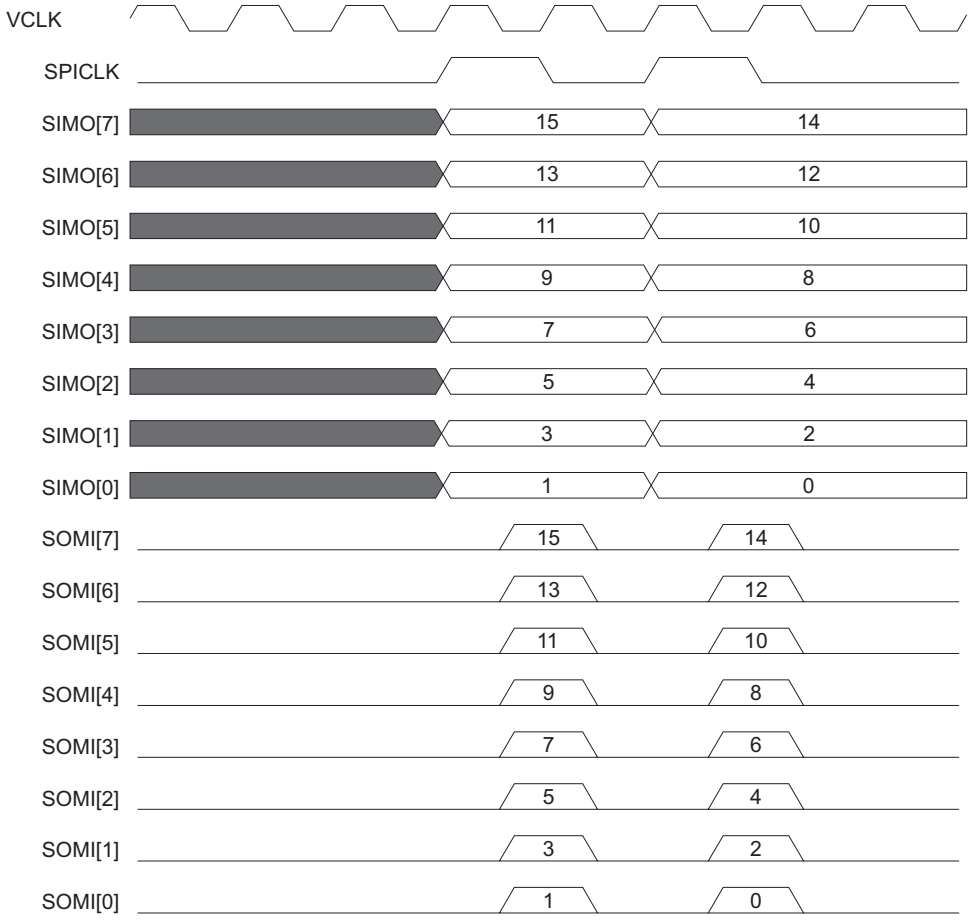
Figure 23-28. Eight-data Line Mode (Phase 0, Polarity 0)



NOTE: Parity Support

Using the parity support in parallel mode may seriously affect throughput. For an eight-line mode to transfer 16 bits of data, only two SPICLK pulses are enough. If parity is enabled, one extra SPICLK pulse will be used to transfer and receive the parity bit. Parity will be transmitted and received on the 0th line regardless of 1/2/4/8-line modes. During the parity bit transfer, other data bits are not valid.

Figure 23-29. 8 Pins Parallel Mode Timing Diagram (Phase 0, Polarity 0)



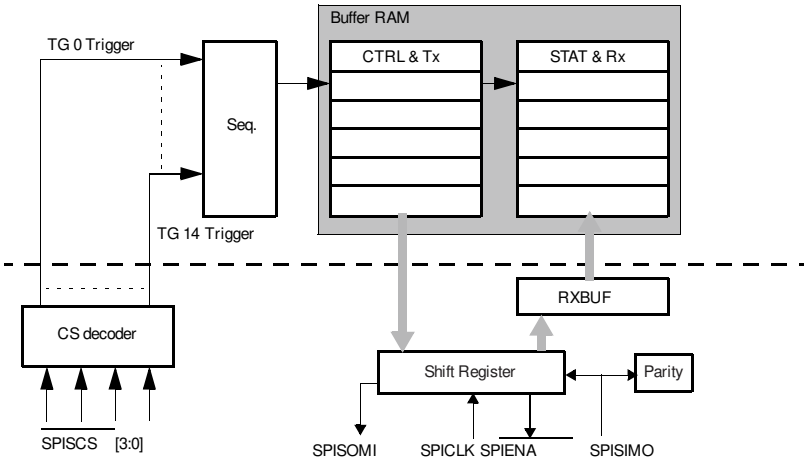
NOTE: Modulo Count Parallel Mode is not supported in this device.

23.2.6.7 MibSPI Slave in Multi-buffer Configuration

When operating in slave mode, the MibSPI uses the chip-select pins 0 to 3 to generate a trigger to the corresponding Transfer Group. For example, putting “0000” on the chip-select pins triggers Transfer Groups 0 and putting “0001” triggers TG 1. When the value “1111” is set to the chip-select, the MibSPI is deselected, that is Transfer Group 15 is not available in slave mode. The chip-select pins 4 to 7 should stay in GPIO mode. In slave mode, the fields like trigger source and trigger event are not taken into account by the sequencer. Only the SPISCS pins can trigger a Transfer Group. The chip-select trigger operates as a level-sensitive trigger. However, when the MibSPI is in 3-pin or 4-pin with SPIENA mode, just one Transfer Group can be triggered and it is restricted to Transfer Group 0 (TG 0). In Slave mode, the PRST field should be cleared to 0. If the corresponding Transfer Group is enabled, the Multi-buffer reads the current buffer of the TG and writes it into SPIDAT1. If Transfer Group is disabled, the Multi-buffer does not update the SPIDAT1 register.

NOTE: If the selected Transfer Group is disabled and no update of the SPIDAT1 register has been done, the data to be transferred is meaningless. Even the received data will not be copied to the multi-buffer RAM. However it will be available on SPIBUF register until it is overwritten by the subsequent receive data.

Figure 23-30. Multi-buffer in Slave Mode



When the SPIDAT1 register is updated, the enable signal is released, and the transaction could begin. If the enable signal is not used, the master should wait for 6 VBUSPCLK cycles before sending the clock to begin the transaction. This time allows the MibSPI to update the SPIDAT1 register.

Once the transaction is finished, the MibSPI writes back the content of the shift-register into the Rx buffer and updates the status field.

NOTE: If all the Transfer Groups are not needed, the number of SPISCS that need to be in functional mode could be reduced to 3, 2 or 1 by using the SPIPC0 register. In these cases, the maximum number of Transfer Group accessible are respectively 7, 3 and 1. The pins that are set in GPIO mode are not decoded.

MibSPI in 3-pin and 4pin (with SPIENA) configuration, supports multi-buffer mode too. However it is restricted to have just one transfer group. Only the transfer group0 (TG0) can be used in this mode. Entire multi-buffer RAM can be configured for TG0 alone. “PSTART” field in TG1CTRL register should be used to configure the size of the multi-buffer (end of the buffers) for TG0.

NOTE: The maximum input frequency on the SPICLK pin when in slave mode is VBUSPCLK frequency /2. If the Slave is configured in either 3-pin or 4-pin (without SPIENA) modes, then, between end of last SPICLK and the start of SPICLK for next buffer, there should be at least 6 VCLK cycles of delay.

23.2.6.8 Transfer Groups

The size of the multi-buffer RAM depends on the implementation. It comprises 0 to 128/256 buffers, whereas 0 buffers considers the special case of no multi-buffer RAM. Each entry in the multi-buffer RAM consists of 4 parts: a 16-bit transmit field, a 16-bit receive field, a 16-bit control field and a 16-bit status field. The multi-buffer RAM can be partitioned into multiple transfer group with variable number of buffers each.

23.2.6.8.1 Configuring Transfer Groups and Trigger Events

Each TG can be configured via one dedicated control register, TGxCTRL. This register even configures the trigger events for the transfer group. The register is described in . The actual number of available control registers varies by device.

23.2.6.8.2 Sequencer-Which Handled the Sequencing of Triggered Transfer Groups

Sequencer(FSM) controls the data flow from the multi-buffer RAM to the Shift Register. The Multi-buffer Control Logic has arbitration logic between VBUS and the Sequencer accessing the multi-buffer RAM. Sequencer picks up a highest priority Transfer Group from among the active TGs to be serviced. For the selected TG the starting buffer to be transferred is obtained from the PSTART of the respective TGxCTRL register.

Sequencer requests for the selected buffer through the Multi-buffer Control Logic, and once it receives the data, it reads the control fields to determine the subsequent action. Once the buffer is determined to be ready for transfer, the data is written to the TX SHIFT REGISTER by the Sequencer. This triggers the Kernel FSM to initiate the SPI transfer.

23.2.6.8.3 Inter-group Prioritization and Arbitration

Transfer Group0 (TG0) has the highest priority and TG15 has the lowest priority among the transfer groups TG0 to TG15.Where as under the following conditions under the following conditions a lower priority Transfer Group cannot be interrupted by a higher priority TG.

- When there's a CSHOLD or LOCK buffer, until the completion of the next buffer transfer which is a non-CSHOLD or non-LOCK buffer, the Transfer Group cannot be interrupted by any higher priority TGs.
- An entire sequence of buffer transfer for NOBRK DMA buffer cannot be interrupted by any higher priority TG.
- Once the last buffer in a Transfer Group is prefetched, a higher priority TG cannot interrupt it until the completion of the Transfer Group.

These prioritizations made among the transfer groups also decide the arbitration logic among the multiple transfer groups which are active

23.2.6.8.4 Transmission Lock Capability

Some slave devices require to have "command" followed by "data". In this case the SPI transaction should not be interrupted by another group transfer. The LOCK bit within each buffer allows consecutive transfer to happen without being interrupted by another higher priority group transfer.

23.2.7 General-Purpose I/O

All of the SPI pins may be programmed via the SPIPCx control registers to be either functional or general-purpose I/O pins.

If the SPI function is to be used, application software must ensure that at least the SPICLK pin and the SOMI and/or SIMO pins are configured as SPI functional pins, and not as GIO pins, or else the SPI state machine will be held in reset, preventing SPI transactions.

SPI pins support:

- internal pull-up resistors
- internal pull-down resistors
- open-drain or push-pull mode
- input-buffer enabling/disabling (controlled by the PULDIS and PSEL bits)

23.2.8 Low-Power Mode

The SPI can be put into either local or global low-power mode. Global low-power mode is asserted by the system and is not controlled by the SPI. During global low-power mode, all clocks to the SPI are turned off, making the module completely inactive.

Local low-power mode is asserted by setting the POWERDOWN (SPIGCR1[8]) bit; setting this bit stops the clocks to the SPI internal logic and registers. Setting the POWERDOWN bit causes the SPI to enter local low-power mode and clearing the POWERDOWN bit causes SPI to exit from local low-power mode. All registers remain accessible during local power-down mode, since the clock to the SPI registers is temporarily re-enabled for each access. RAM buffers are also accessible during low power mode.

NOTE: Since entering a low-power mode has the effect of suspending all state-machine activities, care must be taken when entering such modes to ensure that a valid state is entered when low-power mode is active. Application software must ensure that a low power mode is not entered during a data transfer.

23.2.9 Safety Features

23.2.9.1 Detection of Slave Desynchronization (Master Only)

When a slave supports generation of an enable signal (ENA), desynchronization can be detected. With the enable signal a slave indicates to the master that it is ready to exchange data. A desynchronization can occur if one or more clock edges are missed by the slave. In this case the slave may block the SOMI line until it detects clock edges corresponding to the next data word. This would corrupt the data word of the desynchronized slave and the consecutive data word. A configurable 8-bit time-out counter (T2EDELAY), which is clocked with SPICLK, is implemented to detect this slave malfunction. After the transmission has finished (end of last bit transferred: either last data bit or parity bit) the counter is started. If the ENA signal generated by the slave does not become inactive before the counter overflows, the DESYNC flag is set and an interrupt is asserted (if enabled).

NOTE: Inconsistency of Desynchronization Flag in Compatibility Mode MibSPI

Because of the nature of this error, under some circumstances it is possible for a desync error detected for the previous buffer to be visible in the current buffer. This is due to the fact that receive completion flag/interrupt will be generated when the buffer transfer is completed. But desync will be detected after the buffer transfer is completed. So, if VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct desynchronization condition. This inconsistency in the desync flag is valid only in compatibility mode of MibSPI. In multi-buffer mode, the desync flag is always assured to be for the current buffer.

23.2.9.2 ENA Signal Time-Out (Master Only)

The SPI in master mode waits for the hardware handshake signal (ENA) coming from the addressed slave before performing a data transfer. To avoid stalling the SPI by a non-responsive slave device, a time-out value can be configured using C2EDELAY. If the time-out counter overflows before an active ENA signal is sampled, the TIMEOUT flag in the status register SPIFLG is set and the TIMEOUT flag in the status field of the corresponding buffer is set.

NOTE: When the chip select signal becomes active, no breaks in transmission are allowed. The next arbitration is performed while waiting for the time-out to occur.

23.2.9.3 Data-Length Error

A SPI can generate an error flag by detecting any mismatch in length of received or transmitted data and the programmed character length under certain conditions.

Data-Length Error in Master Mode: During a data transfer, if the SPI detects a de-assertion of the $\overline{\text{SPIENA}}$ pin (by the slave) while the character counter is not overflowed, then an error flag is set to indicate a data-length error. This can be caused by a slave receiving extra clocks (for example, due to noise on the SPICLK line).

NOTE: In a master mode SPI, the data length error will be generated only if the $\overline{\text{SPIENA}}$ pin is enabled as a functional pin.

Data-Length Error in Slave Mode: During a transfer, if the SPI detects a de-assertion of the $\overline{\text{SPISCS}}$ pin before its character length counter overflows, then an error flag is set to indicate a data-length error. This situation can arise If the slave SPI misses one or more SPICLK pulses from the master. This error in slave mode implies that both the transmitted and received data were not complete.

NOTE: In a slave-mode SPI, the data-length error flag will be generated only if at least one of the $\overline{\text{SPISCS}}$(x) pins are configured as functional, and are being used for selecting the slave.

23.2.9.4 Continuous Self-Test (Master/Slave)

During data transfer, the SPI compares its own internal transmit data with its transmit data on the bus. The sample point for the compare is at one-half SPI clock after transmit point. If the data on the bus does not match the expected value, the bit-error (BITERR) flag is set and an interrupt is asserted if enabled.

NOTE: The compare is made from the output pin using its input buffer.

23.2.10 Test Features

23.2.10.1 Internal Loop-Back Test Mode (Master Only)

The internal loop-back self-test mode can be utilized to test the SPI transmit and receive paths, including the shift registers, the SPI buffer registers, and the parity generator. In this mode the transmit signal is internally fedback to the receiver, whereas the SIMO, SOMI, and CLK pin are disconnected; that is, the transmitted data is internally transferred to the corresponding receive buffer while external signals remain unchanged.

This mode allows the CPU to write into the transmit buffer, and check that the receive buffer contains the correct transmit data. If an error occurs the corresponding error is set within the status field.

NOTE: This mode cannot be changed during transmission.

23.2.10.2 Input/Output Loopback Test Mode

Input/Output Loopback Test mode supports the testing of all Input/Output pins without the aid of an external interface. Loopback can be configured as either analog-loopback (loopback through the pin-level input/output buffers) or digital loopback (internal to the SPI module). With Input/Output Loopback, all functional features of the SPI can be tested. Transmit data is fed back through the receive-data line(s). See Figure 23-31 for a diagram of the types of feedback available. The IOLPBKTSTCR register defines all of the available control fields.

In loopback mode, it is also possible to induce various error conditions. See for details of the register field controlling these features.

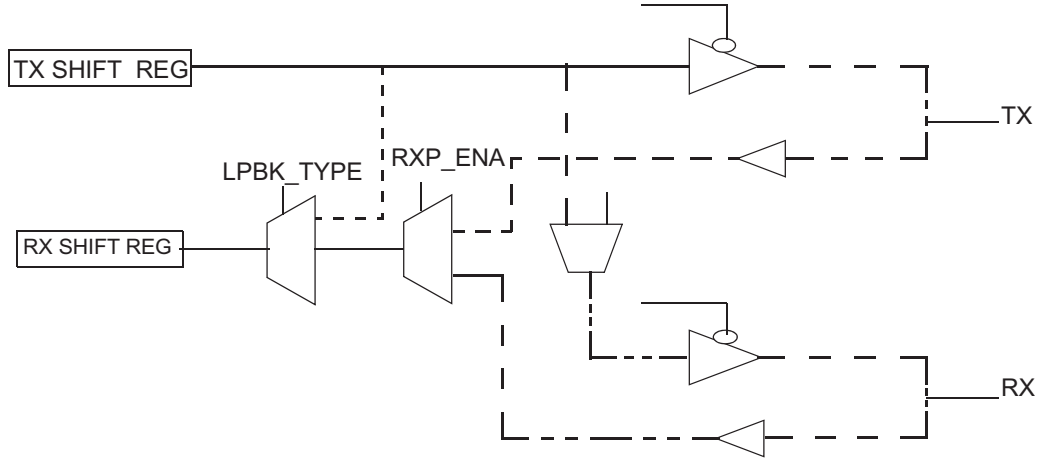
In Input/Output loopback test modes, even when the module is in slave mode, the SPICLK is generated internally. This SPICLK is used for all loopback-mode SPI transactions. Slave-mode features can be tested without the help of another master SPI, using the internally-generated SPICLK. Chip selects are also generated by the slave itself while it is in Input/Output loopback mode.

In Input/Output loopback test modes, if the module is in master mode, the nENA signal is also generated by internal logic so that an external interface is not required.

NOTE: Usage Guideline for Input/Output Loopback

Input/Output Loopback mode should be used with caution because, in some configurations, even the receive pins will be driven with transmit data. During testing, it should be ensured that none of the SPI pins are driven by any other device connected to them. Otherwise, if analog loopback is selected in I/O Loopback mode, then testing may damage the device.

Figure 23-31. I/O Paths During I/O Loopback Modes



1 This diagram is intended to illustrate loopback paths and therefore may omit some normal-mode paths.

23.2.10.2.1 IO Loopback Mode Operation in Slave Mode

In multi-buffer slave mode, there are some additional requirements for using I/O loopback mode (IOLPBK). In multi-buffer slave mode, the chip-select pins are the triggers for various TGs. Enabling the IOLPBK mode by writing 0xA to the IOLPBTSTENA bits of the IOLPBKTSTCR register triggers TG0 by driving $\overline{\text{SPISCS}}$[3:0] to 0x0. The actual number of chip selects can be programmed to have any or all of the $\overline{\text{SPISCS}}$ pins as functional. All other configurations should be completed before enabling the IOLPBK mode in multi-buffer slave mode since it triggers TG0.

After the first buffer transfer is completed, the CSNR[3:0] field of the current buffer is used to trigger the next buffer. So, if multiple TGs are desired to be tested, then the CSNR field of the final buffer in each TG should hold the number of the next TG to be triggered. As long as TG boundaries are well defined and are enabled, the completion of one TG will trigger the next TG.

To stop the transfer in multi-buffer slave mode in I/O Loopback configuration, either IOLPBK mode can be disabled by writing 0x5 to the IOLPBTSTENA bits or all of the TGs can be disabled.

23.2.11 Module Configuration

MibSPI/MibSPIP can be configured to function as Normal SPI and Multi-buffered SPI. Upon power-up or a system-level reset, each bit in the module registers is set to a default state. The registers are writable only after the RESET bit is set to 1.

23.2.11.1 Compatibility (SPI) Mode Configuration

The following list details the configuration steps that software should perform prior to the transmission or reception of data. As long as SPIENA is held low the entire time that the SPI is being configured, the order in which the registers are programmed is not important.

- Enable SPI by setting RESET bit.
- Configure the SIMO, SOMI, CLK and optional CSx, ENA pins for SPI functionality by setting the corresponding bit in SPIPC0 register.
- Configure the module to function as Master or Slave using CLKMOD and MASTER bits.
- Configure the required SPI data format using SPIFMTx register.
- If the module is selected to function as Master, the delay parameters can be configured using SPIDELAY register.
- Enable the Interrupts using SPIINT0 register if required.
- Select the CS to be used by setting CSNR bits in SPIDAT1 register.
- Configure CSHOLD and WDEL bits in SPIDAT1 register if required.
- Select the Data word format by setting DFSEL bits. Select the Number of the configured SPIFMTx register (0 to 3) to used for the communication.
- Set LOOPBACK bit to connect the transmitter to the receiver internally. (This feature is used to perform a self-test. Do not configure for normal communication to external devices).
- Set SPIENA to 1 after the SPI is configured.
- Perform Transmit and receive data, using SPIDAT1 and SPIBUF register.
- User must wait for TXFULL to reset or TXINT before writting next data to SPIDAT1 register.
- User must wait for RXEMPTY to reset or RXINT before reading the data from SPIBUF register.

23.2.11.2 MibSPI Mode Configuration

The following list details the configuration steps that software should perform prior to the transmission or reception of data in MIBSPI mode. As long as SPIENA is held low the entire time that the SPI is being configured, the order in which the registers are programmed is not important.

- Enable SPI by setting RESET bit.
- Set MSPIENA to 1 to get access to multi-buffer mode registers.
- Configure the SIMO, SOMI, CLK and optional CSx, ENA pins for SPI functionality by setting the corresponding bit in SPIPC0 register.
- Configure the module to function as Master or Slave using CLKMOD and MASTER bits.
- Configure the required SPI data format using SPIFMTx register.
- If the module is selected to function as Master, the delay parameters can be configured using SPIDELAY register.
- Check for BUFINITACTIVE bit to be active before configuring MIBSPI RAM. (From Device Power On it take Number of Buffers × Peripheral clock period to initialize complete RAM.)
- Enable the Transfer Group interrupts using TGITENST register if required.
- Enable error interrupts using SPIINT0 register if required.
- Set SPIENA to 1 after the SPI is configured.
- The Trigger Source, Trigger Event, Transfer Group start address for the corresponding Transfer groups can be configured using the corresponding TGxCTRL register.
- Configure LPEND to specify the end address of the last TG.
- Similar to SPIDAT1 register 16 Bit Control fields in every TXRAM buffer in the TG has to be configured.
- Configure one of the eight BUFMODE available for each buffer.
- Fill the data to be transmitted in TXDATA field in TXRAM buffers.
- Configure TGENA bit to enable the required Transfer groups. (In case of Trigger event always setting TGENA will trigger the transfer group).
- At the occurrence of the correct trigger event the Transfer group will be triggered and data gets transmitted and received one after the other with out any CPU intervention.
- User can poll Transfer-group interrupt flag or wait for a transfer-completed interrupt to read and write new data to the buffers.

23.3 Control Registers

This section describes the SPI control, data, and pin registers. The registers support 8-bit, 16-bit and 32-bit writes. The offset is relative to the associated base address of this module in a system. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

NOTE: TI highly recommends that write values corresponding to the reserved locations of registers be maintained as 0 consistently. This allows future enhancements to use these reserved bits as control bits without affecting the functionality of the module with any older versions of software.

23.3.1 MSS_MIBSPIA Registers

Table 23-10 lists the memory-mapped registers for the MSS_MIBSPIA. All register offset addresses not listed in Table 23-10 should be considered as reserved locations and the register contents should not be modified.

Table 23-10. MSS_MIBSPIA Registers

Offset	Acronym	Register Name	Section
0h	SPIGCR0	SPI / MibSPI Global Control Register 0	Section 23.3.2.1
4h	SPIGCR1	SPI / MibSPI Global control register 1	Section 23.3.2.2
8h	SPIINT0	SPI / MibSPI Interrupt Enable Register	Section 23.3.2.3
Ch	SPILVL	SPI / MibSPI Interrupt Level Register	Section 23.3.2.4
10h	SPIFLG	SPI / MibSPI Flag Register	Section 23.3.2.5
14h	SPIPC0	SPI / MibSPI Pin Control Register 0 (SPIPC0) - SPIFUN Note: Duplicate Control Bits for SIMO0 & SOMI0 Bit 24 is not physically implemented. it is a mirror of Bit11. Any write to Bit 24 will be reflected on Bit11 and when Bit 24 & Bit 11 simultaneously written, the value of Bit11 will control the SOMI pin. Read value of Bit 24 always reflects the Bit 11 value. This is true for the Bit 24 & Bit 11 of all of SPIPC0 to SPIPC9 registers. Same is true for SIMO pin with Bit16 & Bit 10 of SPIPC0 to SPIPC9 registers.	Section 23.3.2.6
18h	SPIPC1	SPI / MibSPI Pin Control Register 1 (SPIPC1) - SPIDIR	Section 23.3.2.7
1Ch	SPIPC2	SPI / MibSPI Pin Control Register 2 (SPIPC2) - SPIDIN	Section 23.3.2.8
20h	SPIPC3	SPI / MibSPI Pin Control Register 3 (SPIPC3) - SPIDOUT	Section 23.3.2.9
24h	SPIPC4	SPI / MibSPI Pin Control Register 4 (SPIPC4) - SPIDSET	Section 23.3.2.10
28h	SPIPC5	SPI / MibSPI Pin Control Register 5 (SPIPC5) - SPIDCLR	Section 23.3.2.11
2Ch	SPIPC6	SPI / MibSPI Pin Control Register 6 (SPIPC6) - SPIPDR	Section 23.3.2.12
38h	SPIDAT0	SPI / MibSPI Transmit Data Register 0 Note: Accessibility of SPIDAT0 The SPIDAT0 register is not accessible in Multibuffer Mode of MibSPI. It is only accessible in compatibility mode.	Section 23.3.2.13
3Ch	SPIDAT1	SPI / MibSPI Transmit Data Register 1 When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned.	Section 23.3.2.14
40h	SPIBUF	SPI / MibSPI Receive Buffer Register	Section 23.3.2.15
44h	SPIEMU	SPI / MibSPI Emulation Register Note: All the fields ot SPIEMU register are Read-Only. Read operation on this register under any mode will not have any impact on the status of this or any other registers.	Section 23.3.2.16
48h	SPIDELAY	SPI / MibSPI Delay Register	Section 23.3.2.17
4Ch	SPIDEF	SPI / MibSPI Default Chip select Register	Section 23.3.2.18
50h	SPIFMT0	SPI / MibSPI Data Format Register 0	Section 23.3.2.19
54h	SPIFMT1	SPI / MibSPI Data Format Register 1	Section 23.3.2.20
58h	SPIFMT2	SPI / MibSPI Data Format Register 2	Section 23.3.2.21
5Ch	SPIFMT3	SPI / MibSPI Data Format Register 3	Section 23.3.2.22
60h	TGINTVECT0	SPI Interrupt Vector Register 0 / MibSPI Transfer Group Interrupt Vector Register 0	Section 23.3.2.23
64h	TGINTVECT1	SPI Interrupt Vector Register 1 / MibSPI Transfer Group Interrupt Vector Register 1	Section 23.3.2.24

Table 23-10. MSS_MIBSPIA Registers (continued)

Offset	Acronym	Register Name	Section
68h	SPIPC9	SPI/MibSPI Pin Control Register 9 (SPIPC9) - SPISRSEL	Section 23.3.2.25
6Ch	SPIPMCTRL	SPI/MibSPI Parallel/Modulo Mode Control Register	Section 23.3.2.26
70h	MIBSPIE	MibSPI Enable Register	Section 23.3.2.27
74h	TGITENST	MibSPI Transfer Group Interrupt Enable Set Register	Section 23.3.2.28
78h	TGITENCR	MibSPI Transfer Group Interrupt Enable Clear Register	Section 23.3.2.29
7Ch	TGITLVST	MibSPI Transfer Group Interrupt Level Set Register	Section 23.3.2.30
80h	TGITLVCR	MibSPI Transfer Group Interrupt Level Clear Register	Section 23.3.2.31
84h	TGINTFLAG	Transfer Group Interrupt Flag Register	Section 23.3.2.32
90h	TICKCNT	Tick Count Register	Section 23.3.2.33
94h	LTGPEND	Last Transfer Group End Pointer	Section 23.3.2.34
98h	TG0CTRL	MibSPI Transfer Group Control Register The number of transfer groups is scalable by design up to a maximum of 16. Depending on the implementation the number of transfer groups and hence the number of transfer group control register may vary. Each transfer group can be configured via one dedicated control register. The register description below shows one exemplary control register(x) which is identical for all transfer groups. E.g. the control register for transfer group 2 is named “TG2CTRL” and is located at address base0+98h+4*2.	Section 23.3.2.35
9Ch	TG1CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.36
A0h	TG2CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.37
A4h	TG3CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.38
A8h	TG4CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.39
ACh	TG5CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.40
B0h	TG6CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.41
B4h	TG7CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.42
D8h	DMA0CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.43
DCh	DMA1CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.44
E0h	DMA2CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.45
E4h	DMA3CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.46
E8h	DMA4CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.47
F8h	ICOUNT0	MibSPI DMAxCOUNT	Section 23.3.2.48
FCh	ICOUNT1	MibSPI DMAxCOUNT	Section 23.3.2.49
100h	ICOUNT2	MibSPI DMAxCOUNT	Section 23.3.2.50
104h	ICOUNT3	MibSPI DMAxCOUNT	Section 23.3.2.51
108h	ICOUNT4	MibSPI DMAxCOUNT	Section 23.3.2.52
118h	DMACNTLEN	DMA LARGE COUNT register	Section 23.3.2.53
120h	PAR_ECC_CTRL	Parity/ECC Control Register	Section 23.3.2.54
124h	PAR_ECC_STAT	Parity/ECC Status Register	Section 23.3.2.55
128h	UERRADDR1	Uncorrectable Parity or double bit ECC error Address Register - RXRAM	Section 23.3.2.56
12Ch	UERRADDR0	Uncorrectable Parity or double bit ECC error address register - TXRAM	Section 23.3.2.57
130h	RXOVRN_BUF_ADDR	Receive RAM Overrun Buffer Address Register	Section 23.3.2.58

Table 23-10. MSS_MIBSPIA Registers (continued)

Offset	Acronym	Register Name	Section
134h	IOLPBKTSTCR	SPI/MibSPI IO Loopback Test Control Register This register controls test mode for I/O pins. It also controls whether loop-back should be digital or analog ones in this test mode. In addition it contains control bits to induce some of the error condition into the module. These are to be used for test purpose only. All the control/status bits in this register are valid only when IO LPBK TST ENA field is set to “1010”.	Section 23.3.2.59
138h	EXTENDED_PRESCALE1	SPI/MibSPI Extended Prescale Register 1 (EXTENDED_PRESCALE1 for SPIFMT0 and SPIFMT1) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This is an extension of SPIFMT0 and SPIFMT1 registers. For example, EPRESCALE_FMT1(7:0) of EXTENDED_PRESCALE1 and PRESCALE1(7:0) of SPIFMT1 register will always reflect the same contents. Similarly EPRESCALE_FMT0(7:0) and PRESCALE0(7:0) of SPIFMT0 reflect the same contents.	Section 23.3.2.60
13Ch	EXTENDED_PRESCALE2	SPI/MibSPI Extended Prescale Register 2 (EXTENDED_PRESCALE2 for SPIFMT2 and SPIFMT3) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This register is an extension of SPIFMT2 and SPIFMT3 registers. For example, EPRESCALE_FMT2(7:0) of EXTENDED_PRESCALE2 and PRESCALE2(7:0) of SPIFMT2 register will always reflect the same contents. Similarly EPRESCALE_FMT3(7:0) and PRESCALE3(7:0) of SPIFMT3 reflect the same contents.	Section 23.3.2.61
140h	ECCDIAG_CTRL	ECC Diagnostic Control register	Section 23.3.2.62
144h	ECCDIAG_STAT	ECC Diagnostic Status register	Section 23.3.2.63
148h	SBERRADDR1	Single Bit Error Address Register - RXRAM	Section 23.3.2.64
14Ch	SBERRADDR0	Single Bit ECC Error Address Register - TXRAM	Section 23.3.2.65
1FCh	SPIREV	SPI / MibSPI Revision ID Register	Section 23.3.2.66

Complex bit access types are encoded to fit into small table cells. Table 23-11 shows the codes that are used for access types in this section.

Table 23-11. MSS_MIBSPIA Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	-0 R	Returns 0s Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

23.3.1.1 SPIGCR0 Register (Offset = 0h) [reset = 0h]

SPIGCR0 is shown in Figure 23-98 and described in Table 23-80.

Return to Summary Table.

SPI / MibSPI Global Control Register 0

Figure 23-32. SPIGCR0 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							nRESET
R-0h							R/W-0h

Table 23-12. SPIGCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	Reserved, Reads return ‘0’ and writes have no effect.
0	nRESET	R/W	0h	This is the local reset control for the module. This bit needs to be set to ‘1’ before any operation on SPI / MibSPI can be done. Only after setting this bit to ‘1’, the Auto Initialization of Multibuffer RAM starts. Clearing this bit to ‘0’ will result in all of the control and status register values to return to their default values. 0 = SPI / MibSPI is in reset state 1 = SPI / MibSPI is out of reset state.

23.3.1.2 SPIGCR1 Register (Offset = 4h) [reset = 0h]

SPIGCR1 is shown in Figure 23-99 and described in Table 23-81.

Return to Summary Table.

SPI / MibSPI Global control register 1

Figure 23-33. SPIGCR1 Register

31	30	29	28	27	26	25	24
NU4							SPIEN
R-0h							R/W-0h
23	22	21	20	19	18	17	16
NU3							LOOPBACK
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU2							POWERDOWN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NU1						CLKMOD	MASTER
R-0h						R/W-0h	R/W-0h

Table 23-13. SPIGCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	SPIEN	R/W	0h	SPI enable. This bit enables the SPI/MibSPI transfers. This bit must be set to 1 after all other SPI / MibSPI configuration bits have been written. When SPIEN bit is 0 or cleared to 0, the following SPI/MibSPI registers get forced to their default states (to ‘0’s except for RXEMPTY bit in SPIBUF): - Both TX & RX Shift Registers - The TXDATA fields of SPIDAT0 and SPIDAT1 registers - All the fields of SPIFLG register - Contents of SPIBUF & the internal RXBUF registers 0=SPI / MibSPI is not activated for transfers. 1=Activates SPI / MibSPI
23-17	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
16	LOOPBACK	R/W	0h	LOOP BACK. Internal loop-back test mode. The internal self-test option can be enabled by setting this bit. If the SPISIMO and SPISOMI pins are configured with SPI functionality, then the SPISIMO pin is internally connected to the SPISOMI pin. The transmit data is looped back as receive data and is stored in the receive field of the concerned buffer. Externally, during loop-back operation, the SPICLK pin outputs an inactive value and SPISOMI remains in high-impedance state. The SPI / MibSPI has to be initialized in master mode before the loop-back can be selected. If the SPI / MibSPI is initialized in slave mode or a data transfer is ongoing, errors may result. 1 =Internal loop-back test mode enabled. 0 =Internal loop-back test mode disabled. This loopback mode can be used only in Master mode. This automatically selects digital loopback path. When this Loopback mode is selected, CLKMOD bit should be set to ‘1’, meaning that SPICLK can only be internal.
15-9	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
8	POWERDOWN	R/W	0h	POWERDOWN. When active, the SPI / MibSPI state machines enter a powerdown state. 0=MibSPI in active mode 1=MibSPI in powerdown mode
7-2	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
1	CLKMOD	R/W	0h	CLKMOD. Clock mode Selects either an internal or external clock source. This bit also determines the I/O direction of the SPIENA and SPISCS[7:0] pins in functional mode. 0=Clock is external 1=Clock is internal

Table 23-13. SPIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	MASTER	R/W	0h	MASTER: SPISIMO/SPISOMI pin direction determination. Determines the direction of the SPISIMO and SPISOMI pins. This bit determines whether the SPI/MibSPI is in Master mode or Slave mode. This bit also controls the Master-only features like the C2T/T2C counters, C2E/T2E timers, most of the Error conditions specific to Master mode. 0=SPISIMO pin an input, SPISOMI pin an output 1=SPISOMI pin an input, SPISIMO pin an output Note: Although there are two different bits which control the Master/Slave mode functions, only two of their combinations are valid. For compatibility reasons both the bits are retained. For Master mode of operation: MASTER = '1', CLKMOD = '1' For Slave mode of operation: MASTER = '0', CLKMOD = '0' Any other combinations of these two bits may not yield any desirable operation of the module.

23.3.1.3 SPIINT0 Register (Offset = 8h) [reset = 0h]

SPIINT0 is shown in Figure 23-100 and described in Table 23-82.

Return to Summary Table.

SPI / MibSPI Interrupt Enable Register

Figure 23-34. SPIINT0 Register

31	30	29	28	27	26	25	24
NU5							ENABLEHIGHZ
R-0h							R/W-0h
23	22	21	20	19	18	17	16
NU4							DMAREQEN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU3						TXINTENA	RXINTENA
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
NU2	OVRNINTENA	NU1	BITERRENA	DESYNCENA	PARERRENA	TIMEOUTENA	DLENERRENA
R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-14. SPIINT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU5	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	ENABLEHIGHZ	R/W	0h	SPIENA pin high-z enable. When active, the SPIENA pin (when it is configured as a WAIT functional output signal in a slave SPI) is forced to place it is output in high-z when not driving a low signal. If inactive, then the pin will output both a high and a low signal. 0=SPIENA pin is pulled high when not active. 1=SPIENA pin remains in high-z when not active.
23-17	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
16	DMAREQEN	R/W	0h	DMA request enable. Enables the DMA request signal to be generated for both receive and transmit channels. Enable DMA REQ only after setting the SPIEN bit to ‘1’. 0=DMA is not used 1=DMA Requests will be generated. A DMA request will be generated on TX DMA REQ line each time a transmit data is copied to the Shift Register either from TXBUF or directly from SPIDAT0/SPIDAT1 writes. A DMA request will be generated on RX DMA REQ line each time a received data is copied to SPIBUF register either from RXBUF or directly from the Shift Register.
15-10	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
9	TXINTENA	R/W	0h	An interrupt is to be generated everytime data is written to the “Shift Register”, so that a new data can be written to TXBUF. Setting this bit will generate an interrupt if TXINTFLG bit (SPIFLG.9) is set to ‘1’. 0=No interrupt will be generated upon TXINTFLG getting set to ‘1’. 1=Interrupt will be generated upon TXINTFLG getting set to ‘1’. An interrupt request will be generated as soon as this bit is set to ‘1’. By default it will be generated on INT0 line. SPILVL register can be programmed before-hand to change this default.
8	RXINTENA	R/W	0h	An interrupt is to be generated when the RXINTFLAG bit (SPIFLG.8) is set by hardware. Otherwise, no interrupt will be generated. 0=Interrupt will not be generated 1=Interrupt will be generated Both Transmitter Empty & Receiver Full interrupts are valid in SPI or Compatibility mode of MibSPI only. In Multibuffered mode these interrupts will not be generated even if enabled.
7	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.

Table 23-14. SPIINT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	OVRNINTENA	R/W	0h	Overrun interrupt enable. An interrupt is to be generated when the RCVR OVRN flag bit (SPIFLG.6) is set by hardware. Otherwise, no interrupt will be generated. 0=Overrun interrupt will not be generated 1=Overrun interrupt will be generated
5	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
4	BITERRENA	R/W	0h	Enables interrupt on bit error. 1 =Enables an interrupt on a bit error (BITERR = 1). 0 =No interrupt asserted upon bit error.
3	DESYNCENA	R/W	0h	Enables interrupt on de-synchronized slave. DESYNCENA is used in master mode only. 1 =Enables an interrupt on de-synchronization of the slave (DESYNC = 1). 0 =No interrupt asserted upon de-synchronization error.
2	PARERRENA	R/W	0h	Enables interrupt on parity error. 1 =Enables an interrupt on a parity error (PARITYERR = 1). 0 =No interrupt asserted upon parity error.
1	TIMEOUTENA	R/W	0h	Enables interrupt on ENA signal time-out. 1 =Enables an interrupt on a time-out of the ENA signal (TIMEOUT = 1). 0 =No interrupt asserted upon ENA signal time-out.
0	DLENERRENA	R/W	0h	Data Length Error interrupt Enable. 1 = Enables an interrupt when Data Length Error occurs. 0 = No interrupt is generated upon Data Length Error. A Data Length Error occurs under the following conditions. Master: In a 4-pin with SPIENA mode or 5-pin mode, if the SPIENA pin from the slave is deasserted before the Master has completed its transfer, the Data Length Error is set. That is, if the Character Length counter has not completed counting while SPIENA deassertion is detected, then it means that the Slave has neither received full data from the Master nor has it transmitted complete data. Slave: In a 4-pin with ChipSelects mode or 5-pin mode, if the incoming valid SPISCS pin is de-activated before the Character Length counter completes counting, then Data Length Error is set.

23.3.1.4 SPILVL Register (Offset = Ch) [reset = 0h]

SPILVL is shown in Figure 23-101 and described in Table 23-83.

Return to Summary Table.

SPI / MibSPI Interrupt Level Register

Figure 23-35. SPILVL Register

31	30	29	28	27	26	25	24
NU3							
R-0h							
23	22	21	20	19	18	17	16
NU3							
R-0h							
15	14	13	12	11	10	9	8
NU3						TXINTLVL	RXINTLVL
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
NU2	OVRNINTLVL	NU1	BITERRLVL	DESYNCLVL	PARERRLVL	TIMEOUTLVL	DLENERRLVL
R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-15. SPILVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
9	TXINTLVL	R/W	0h	Transmit Interrupt Level. 1 =Transmit interrupt is mapped to interrupt line INT1. 0 =Transmit interrupt is mapped to interrupt line INT0.
8	RXINTLVL	R/W	0h	Receive interrupt level. 1 =Receive interrupt is mapped to interrupt line INT1. 0 =Receive interrupt is mapped to interrupt line INT0.
7	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
6	OVRNINTLVL	R/W	0h	Receive Overrun interrupt level. 1 =Receive Overrun interrupt is mapped to interrupt line INT1. 0 =Receive Overrun interrupt is mapped to interrupt line INT0.
5	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
4	BITERRLVL	R/W	0h	Bit error interrupt level. 1 =bit error interrupt is mapped to interrupt line INT1. 0 =bit error interrupt is mapped to interrupt line INT0.
3	DESYNCLVL	R/W	0h	De-synchronized slave interrupt level. DESYNCLVL is used in master mode only. 1 =An interrupt due to de-synchronization of the slave (DESYNC = 1) is mapped to interrupt line INT1. 0 =An interrupt due to de-synchronization of the slave (DESYNC = 1) is mapped to interrupt line INT0.
2	PARERRLVL	R/W	0h	Parity error interrupt level. 1 =A parity error interrupt (PARITYERR = 1) is mapped to interrupt line INT1. 0 =A parity error interrupt (PARITYERR = 1) is mapped to interrupt line INT0.
1	TIMEOUTLVL	R/W	0h	SPIENA pin Time-out interrupt level. 1 =An interrupt on a time-out of the ENA signal (TIMEOUT = 1) is mapped to interrupt line INT1. 0 =An interrupt on a time-out of the ENA signal (TIMEOUT = 1) is mapped to interrupt line INT0.
0	DLENERRLVL	R/W	0h	Data Length Error interrupt Enable Level. 1 = An interrupt on Data Length Error is mapped to interrupt line INT1. 0 = An interrupt on Data Length Error is mapped to interrupt line INT0.

23.3.1.5 SPIFLG Register (Offset = 10h) [reset = 0h]

SPIFLG is shown in Figure 23-102 and described in Table 23-84.

Return to Summary Table.

SPI / MibSPI Flag Register

Figure 23-36. SPIFLG Register

31	30	29	28	27	26	25	24
NU4							BUFINITACTIVE
R-0h							R-0h
23	22	21	20	19	18	17	16
NU3							
R-0h							
15	14	13	12	11	10	9	8
NU3						TXINTFLG	RXINTFLG
R-0h						R-0h	0h
7	6	5	4	3	2	1	0
NU2	OVRNINTFLG	NU1	BITERRFLG	DESYNCFLG	PARERRFLG	TIMEOUTFLG	DLENERRFLG
R-0h	0h	R-0h	0h	0h	0h	0h	0h

Table 23-16. SPIFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	BUFINITACTIVE	R	0h	Indicates the status of Multibuffer initialization process. Software should poll this bit to determine if it can proceed with the configuration of Multibuffer mode registers or Multibuffer RAM handling. Refer to Section 3.10.7 for details on Initialization of Multibuffer RAM. 1 = Multibuffer RAM is still being initialized. Do not attempt to write to either Multibuffer RAM or any Multibuffer mode registers. Refer to Section 3.3 for a classification of registers into compatibility mode and Multibuffer mode. 0 = Multibuffer RAM initialization is complete. This bit will show a value of ‘1’ as long as the nRESET bit is ‘0’, but does not really indicate that Buffer initialization is underway. Internal automatic buffer initialization starts only when the nRESET bit is set to ‘1’. For SPI, this bit reads ‘1’ always. For MibSPI, BUF INIT ACTIVE bit will show up as ‘1’ for a maximum of 128/256 (will vary depending upon the actual size of the Multibuffer RAM implemented) VCLK cycles after the nRESET bit in GCR0 is set to ‘1’ and then settle to ‘0’. If Auto Memory Initialization is triggered through System (MEM_AUTO_INIT pulse), then BUF INIT ACTIVE bit will show up as ‘1’ for a maximum of 128/256 (will vary depending upon the actual size of the Multibuffer RAM implemented) VCLK cycles and then settle to ‘0’.
23-10	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
9	TXINTFLG	R	0h	Transmitter Empty Interrupt Flag. Serves as an interrupt flag indicating that Transmit Buffer (TXBUF) is empty and a new data can be written to it. This flag is set when a data is copied to the “Shift Register” either directly or from the TXBUF register. This bit is cleared by one of following ways: Writing a new data to either SPIDAT0 or SPIDAT1 Writing a ‘0’ to SPIEN (SPIGCR1.24) 0= Transmit Buffer is now full. No interrupt pending for Transmitter Empty 1= Transmit Buffer is empty. An interrupt is pending to fill the transmitter.

Table 23-16. SPIFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RXINTFLG		0h	Receiver Full Interrupt Flag. This flag is set when a word is received and copied into the buffer register (SPIBUF). If RXINTEN is enabled, an interrupt is also generated. During emulation mode, however, a read to the emulation register (SPIEMU) does not clear this flag bit. This bit is cleared under the following ways: Reading the SPIBUF register Reading TGINTVECT0 or TGINTVECT1 register when there is a “Receive Buffer Full” interrupt Writing a ‘1’ to this bit Writing a ‘0’ to SPIEN (SPIGCR1.24) System reset 0= No new received data pending. Receive buffer is Empty. 1= A newly received data is ready to be read. Receive buffer is full. Note: Exception for clearing of RXINT If both SPIBUF and RXBUF (internal buffer) are full, then, reading TGINTVECT0 or TGINTVECT1 register (while it shows 10010) does not clear the RXINTFLG in SPIFLG register. In this case, only way to clear the Interrupt is to read the SPIBUF (twice) and clear all the received data. Note: Side effects of Write Clear to RXINTFLG Clearing RXINTFLG bit by writing a ‘1’ before reading the SPIBUF sets the RXEMPTY bit of the SPIBUF register too. This way, one can ignore a received data. However, if the internal RXBUF is already full, the data from RXBUF will be copied to SPIBUF and RXEMPTY bit will be cleared again. SPIBUF contents should be read first if this situation needs to be avoided.
7	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
6	OVRNINTFLG		0h	Receiver overrun flag. The SPI / MibSPI hardware sets this bit when a receive operation completes before the previous character has been read from the receive buffer. The bit indicates that the last received character has been overwritten and therefore lost. The SPI / MibSPI will generate an interrupt request if this bit is set and the OVRN INTEN bit (SPIINT0.6) is set high. 0 = Overrun condition did not occur 1 = Overrun condition has occurred In SPI or Compatibility mode of MibSPI, this bit is cleared under the following conditions: Reading TGINTVECT0 or TGINTVECT1 register when there is a “Receive Buffer Overrun” interrupt Writing a ‘1’ to OVRNINTFLG in SPIFLG register itself Reading SPIBUF register does not clear this OVRNINTFLG bit. If an RXOVRN interrupt is detected, then the SPIBUF may need to be read twice to get to the Overrun buffer. This is due to the fact that the Overrun will always occur to the internal RXBUF. Each read to the SPIBUF will result in RXBUF contents (if it is full) getting copied to SPIBUF. Note: A special condition under which OVRNINTFLG flag gets set. If both SPIBUF & RXBUF are already full and while another buffer receive is underway, if any errors like TIMEOUT, BITERR & DLEN_ERR occur, then RXOVR in RXBUF & OVRNINTFLG in SPIFLG registers will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a normal Receiver Overrun. In Multibuffer mode of MibSPI, this bit is cleared under the following conditions. Reading the RXOVRN_BUF_ADDR register Writing a ‘1’ to OVRNINTFLG in SPIFLG register itself In Multibuffer mode, if OVRNINTFLG is set, then the address of the buffer which experienced the Overrun is available in RXOVRN_BUF_ADDR.
5	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
4	BITERRFLG		0h	Mismatch of internal transmit data and transmitted data. 1 =A bit error occurred. The SPI / MibSPI samples the signal of the transmit pin (master: SIMO, slave: SOMI) at the receive point (half clock cycle after transmit point). If the sampled value differs from the transmitted value a bit error is detected and the Flag BITERR is set. If BITERRENA is set an interrupt is asserted. A possible reason for a bit error can be a to high bit rate / capacitive load or another master/slave trying to transmit at the same time. 0 =No bit error occurred. This flag can be cleared by one of the following ways. Write a ‘1’ to this bit. Set SPIEN bit to ‘0’.

Table 23-16. SPIFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	DESYNCFLG		0h	De-synchronization of slave device. De-synchronization monitor is active in master mode only. 1 = A slave device is de-synchronized. The master monitors the ENAble signal coming from the slave device and sets the DESYNC flag after the last bit is transmitted plus tT2EDELAY (see Section 8.21). If DESYNCENA is set an interrupt is asserted. De-synchronization can occur if a slave device misses a clock edge coming from the master. 0 =No slave de-synchronization detected. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'. Note: Inconsistency of Desync flag in SPI/Compatibility mode MibSPI Due to the nature of this Error, under some circumstances it is possible for Desync error detected for the previous buffer to be visible in the current buffer. This is due to the fact that Receive Completion flag/interrupt will be generated when the buffer transfer is completed. But Desync will be detected after the buffer transfer is completed. So, if VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct Desync condition. This inconsistency in Desync flag is valid only in SPI or Compatibility mode of MibSPI. In Multibuffer mode, Desync flag is always guaranteed to be for the current buffer.
2	PARERRFLG		0h	Calculated parity differs from received parity bit. 1 =A parity error occurred. If the parity generator is enabled (can be selected individually for each buffer) an even or odd parity bit is added at the end of a data word (see Section 8.23). During reception of the data word the parity generator calculates the reference parity and compares it to the received parity bit. In the event of a mismatch the PARITYERR flag is set and an interrupt is asserted if PARERRENA is set. 0 =No parity error detected. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'.
1	TIMEOUTFLG		0h	Time-out due to non-activation of ENA signal. 1 =An ENA signal time-out occurred. The SPI / MibSPI generates a time-out because the slave hasn't responded in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition the TIMOUT flag in the status field of the corresponding buffer is set. The transmit request of the concerned buffer is cleared, i.e. the SPI / MibSPI doesn't re-start a data transfer from this buffer. 0 =No ENA-signal time-out occurred. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'.
0	DLENERRFLG		0h	Data Length Error Flag. 1 = A Data Length Error has occured. 0 = No Data Length Error has occured. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'. A Data Length Error occurs under the following conditions. Master: In a 4-pin with SPIENA mode or 5-pin mode, if the SPIENA pin from the slave is deasserted before the Master has completed its transfer, the Data Length Error is set. That is, if the Character Length counter has not completed counting while SPIENA pin deassertion is detected, then it means that the Slave has neither received full data from the Master nor has it transmitted complete data. Slave: In a 4-pin with ChipSelects mode or 5-pin mode, if the incoming valid SPISCS pin is de-activated before the Character Length counter completes counting, then Data Length Error is set. Note: Clearing of Transmission Error Flags in SPIBUF during Error conditions Whenever any Transmission Errors (TIMEOUT, BITERR, DLEN_ERR, PARITY_ERR, DESYNC) are detected, and the Error Flag are cleared by writing to the Error bit in SPIFLG register, the corresponding Error flag in SPIBUF does not get cleared. Software needs to read the SPIBUF until it becomes empty before proceeding. This ensures that all the older status bits in SPIBUF are cleared before starting the next transfer.

23.3.1.6 SPIPC0 Register (Offset = 14h) [reset = 0h]

SPIPC0 is shown in Figure 23-103 and described in Table 23-85.

Return to Summary Table.

SPI / MibSPI Pin Control Register 0 (SPIPC0) - SPIFUN Note: Duplicate Control Bits for SIMO0 & SOMI0 Bit 24 is not physically implemented. it is a mirror of Bit11. Any write to Bit 24 will be reflected on Bit11 and when Bit 24 & Bit 11 simultaneously written, the value of Bit11 will control the SOMI pin. Read value of Bit 24 always reflects the Bit 11 value. This is true for the Bit 24 & Bit 11 of all of SPIPC0 to SPIPC9 registers. Same is true for SIMO pin with Bit16 & Bit 10 of SPIPC0 to SPIPC9 registers.

Figure 23-37. SPIPC0 Register

31	30	29	28	27	26	25	24
SOMIFUN							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOFUN							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIFUN0	SIMOFUN0	CLKFUN	ENAFUN
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSFUN							
R/W-0h							

Table 23-17. SPIPC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIFUN	R/W	0h	Slave out, master in function. Determines whether the SPISOMIx pins are to be used as a general-purpose I/O pin or as a SPI / MibSPI functional pin. 0=SPISOMIx pin is a GPIO 1=SPISOMIx pin is a SPI / MibSPI functional pin
23-16	SIMOFUN	R/W	0h	Slave in, master out function. Determines whether the SPISIMOx pin is to be used as a general-purpose I/O pin or as a SPI / MibSPI functional pin. 0=SPISIMOx pin is a GPIO 1=SPISIMOx pin is a SPI / MibSPI functional pin Note: Generic based bit implementation Register bits 31 to 24 and 23 to 16 of SPIPC0 to SPIPC9 are implemented depending upon the generic parameter NUM_PARLL_PINS which determines the number of SIMO/SOMI data lines to be supported. Only if 8 dataline support is selected at the time of logic synthesis, bits 31 to 16 are implemented. Un-implemented bits return ‘0’ upon read and are not writable.
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIFUN0	R/W	0h	Slave out, master in function. Determines whether the SPISOMI0 pin is to be used as a general-purpose I/O pin or as a SPI / MibSPI functional pin. 0=SPISOMI0 pin is a GPIO 1=SPISOMI0 pin is a SPI / MibSPI functional pin Note: Bit 11 or bit 24 can be used to set the function mode for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOFUN0	R/W	0h	Slave in, master out function. Determines whether the SPISIMO0 pin is to be used as a general-purpose I/O pin, or as a SPI / MibSPI functional pin. 0=SPISIMO0 pin is a GPIO 1=SPISIMO0 pin is a SPI / MibSPI functional pin Note: Bit 10 or bit 16 can be used to set the function mode for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKFUN	R/W	0h	SPI / MibSPI clock function. Determines whether the SPICLK pin is to be used as a general-purpose I/O pin, or as a SPI / MibSPI functional pin. 0=SPICLK pin is a GPIO 1=SPICLK pin is a SPI / MibSPI functional pin

Table 23-17. SPIPC0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ENAFUN	R/W	0h	SPIENA function. Determines whether the SPIENA pin is to be used as a general-purpose I/O pin, or as a SPI / MibSPI functional pin. 0=SPIENA pin is a GPIO 1=SPIENA pin is a SPI / MibSPI functional pin
7-0	SCSFUN	R/W	0h	SPISCS[7:0] function. Determines whether the SPISCSx pins are to be used as a general-purpose I/O pins or as SPI functional pins. If the slave SPISCSx pins are in functional mode and receive an inactive high signal, the slave SPI will place it is output in high-z and disable shifting. 0=SPISCSx pin is a GPIO 1=SPISCSx pin is a SPI functional pin Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSFUN[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read ‘0’ always.

23.3.1.7 SPIPC1 Register (Offset = 18h) [reset = 0h]

SPIPC1 is shown in Figure 23-104 and described in Table 23-86.

Return to Summary Table.

SPI / MibSPI Pin Control Register 1 (SPIPC1) - SPIDIR

Figure 23-38. SPIPC1 Register

31	30	29	28	27	26	25	24
SOMIDIR							
R/W-0h							
23	22	21	20	19	18	17	16
SIMODIR							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIDIR0	SIMODIR0	CLKDIR	ENADIR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSDIR							
R/W-0h							

Table 23-18. SPIPC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIDIR	R/W	0h	SPISOMIx direction. Controls the direction of the SPISOMIx pin when it is used as a general-purpose I/O pin. If the SPISOMIx pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISOMIx pin is an input 1=SPISOMIx pin is an output
23-16	SIMODIR	R/W	0h	SPISIMOx direction. Controls the direction of the SPISIMOx pin when it is used as a general-purpose I/O pin. If the SPISIMOx pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISIMOx pin is an input 1=SPISIMOx pin is an output
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIDIR0	R/W	0h	SPISOMI0 direction. Controls the direction of the SPISOMI0 pin when it is used as a general-purpose I/O pin. If the SPISOMI0 pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISOMI0 pin is an input 1=SPISOMI0 pin is an output Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMODIR0	R/W	0h	SPISIMO0 direction. Controls the direction of the SPISIMO0 pin when it is used as a general-purpose I/O pin. If the SPISIMO0 pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISIMO0 pin is an input 1=SPISIMO0 pin is an output Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKDIR	R/W	0h	SPICLK direction. Controls the direction of the SPICLK pin when it is used as a general-purpose I/O pin. In functional mode, the I/O direction is determined by the CLKMOD bit. 0=SPICLK pin is an input 1=SPICLK pin is an output
8	ENADIR	R/W	0h	SPIENA direction. Controls the direction of the SPIENA pin when it is used as a general-purpose I/O. If the SPIENA pin is used as a functional pin, then the I/O direction is determined by the CLKMOD bit (SPIGCR1.1). 0=SPIENA pin is an input 1=SPIENA pin is an output

Table 23-18. SPIPC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	SCSDIR	R/W	0h	SPISCS[7:0] direction. Controls the direction of the SPISCSx pins when they are used as a general-purpose I/O pin. Each pins could be configured independently from the others If the SPISCSx is used as a SPI functional pin, the I/O direction is determined by the CLKMOD bit (SPIGCR1.1). 0=SPISCSx pin is an input 1=SPISCSx pin is an output Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSDIR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always.

23.3.1.8 SPIPC2 Register (Offset = 1Ch) [reset = 0h]

SPIPC2 is shown in Figure 23-105 and described in Table 23-87.

Return to Summary Table.

SPI / MibSPI Pin Control Register 2 (SPIPC2) - SPIDIN

Figure 23-39. SPIPC2 Register

31	30	29	28	27	26	25	24
SOMIDIN							
R-0h							
23	22	21	20	19	18	17	16
SIMODIN							
R-0h							
15	14	13	12	11	10	9	8
NU				SOMIDIN0	SIMODIN0	CLKDIN	ENADIN
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SCSDIN							
R-0h							

Table 23-19. SPIPC2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIDIN	R	0h	SPISOMIx data in. Reflects the value of the SPISOMIx pin. 0=Current value on SPISOMIx pin is logic 0. 1=Current value on SPISOMIx pin is logic 1
23-16	SIMODIN	R	0h	SPISIMOx data in. Reflects the value of the SPISIMOx pin. 0=Current value on SPISIMOx pin is logic 0. 1=Current value on SPISIMOx pin is logic 1
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIDIN0	R	0h	SPISOMI0 data in. Reflects the value of the SPISOMI0 pin. 0=Current value on SPISOMI0 pin is logic 0. 1=Current value on SPISOMI0 pin is logic 1 Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMODIN0	R	0h	SPISIMO0 data in. Reflects the value of the SPISIMO0 pin. 0=Current value on SPISIMO0 pin is logic 0. 1=Current value on SPISIMO0 pin is logic 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKDIN	R	0h	Clock data in. Reflects the value of the SPICLK pin. 0=Current value on SPICLK pin is logic 0. 1=Current value on SPICLK pin is logic 1
8	ENADIN	R	0h	SPIENA data in. Reflects the value of the SPIENA pin. 0=Current value on SPIENA pin is logic 0. 1=Current value on SPIENA pin is logic 1
7-0	SCSDIN	R	0h	SPISCS[7:0] data in. Reflects the value of the SPISCSx pins. 0=Current value on SPISCSx pin is logic 0. 1=Current value on SPISCSx pin is logic 1 Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSDIN[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will read ‘0’ always.

23.3.1.9 SPIPC3 Register (Offset = 20h) [reset = 0h]

SPIPC3 is shown in Figure 23-106 and described in Table 23-88.

Return to Summary Table.

SPI / MibSPI Pin Control Register 3 (SPIPC3) - SPIDOUT

Figure 23-40. SPIPC3 Register

31	30	29	28	27	26	25	24
SOMIDOUT							
R/W-0h							
23	22	21	20	19	18	17	16
SIMODOUT							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIDOUT0	SIMODOUT0	CLKDOUT	ENADOUT
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSDOUT							
R/W-0h							

Table 23-20. SPIPC3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIDOUT	R/W	0h	SPISOMIx dataout write. Only active when the SPISOMIx pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISOMIx pin is logic 0. 1=Current value on SPISOMIx pin is logic 1
23-16	SIMODOUT	R/W	0h	SPISIMOx dataout write. Only active when the SPISIMOx pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISIMOx pin is logic 0. 1=Current value on SPISIMOx pin is logic 1
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIDOUT0	R/W	0h	SPISOMI0 dataout write. Only active when the SPISOMI0 pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISOMI0 pin is logic 0. 1=Current value on SPISOMI0 pin is logic 1. Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMODOUT0	R/W	0h	SPISIMO0 dataout write. Only active when the SPISIMO0 pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISIMO0 pin is logic 0. 1=Current value on SPISIMO0 pin is logic 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKDOUT	R/W	0h	SPICLK dataout write. Only active when the SPICLK pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPICLK pin is logic 0. 1=Current value on SPICLK pin is logic 1
8	ENADOUT	R/W	0h	SPIENA dataout write. Only active when the SPIENA pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPIENA pin is logic 0. 1=Current value on SPIENA pin is logic 1

Table 23-20. SPIPC3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	SCSDOUT	R/W	0h	SPISCS[7:0] dataout write. Only active when the SPISCSx pins are configured as a general-purpose I/O pins and configured as an output pins. The value of these bit indicates the value sent to the pins. 0=Current value on SPISCSx pin is logic 0. 1=Current value on SPISCSx pin is logic 1 Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSDOUT[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always.

23.3.1.10 SPIPC4 Register (Offset = 24h) [reset = 0h]

SPIPC4 is shown in Figure 23-107 and described in Table 23-89.

Return to Summary Table.

SPI / MibSPI Pin Control Register 4 (SPIPC4) - SPIDSET

Figure 23-41. SPIPC4 Register

31	30	29	28	27	26	25	24
SOMISET							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOSET							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMISET0	SIMOSET0	CLKSET	ENASET
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSSET							
R/W-0h							

Table 23-21. SPIPC4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMISET	R/W	0h	SPISOMIx dataout set. Only active when the SPISOMIx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISOMIDOUTx bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISOMIx pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUTx is 0. 1= Current value on SOMIDOUTx is 1.
23-16	SIMOSET	R/W	0h	SPISIMOx dataout set. Only active when the SPISIMOx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISIMODOUTx bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISIMOx pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUTx is 0. 1= Current value on SIMODOUTx is 1
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMISET0	R/W	0h	SPISOMI0 dataout set. Only active when the SPISOMI0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISOMIDOUT bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISOMI0 pin if it is in General Purpose O/P mode Read: 0= Current value on SOMIDOUT0 is 0. 1= Current value on SOMIDOUT0 is 1. Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOSET0	R/W	0h	SPISIMO0 dataout set. Only active when the SPISIMO0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISIMODOUT bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISIMO0 pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUT0 is 0. 1= Current value on SIMODOUT0 is 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKSET	R/W	0h	SPICLK dataout set. Only active when the SPICLK pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding CLKDOUT bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPICLK pin if it is in General Purpose O/P mode Read: 0= Current value on CLKDOUT pin is logic 0. 1= Current value on CLKDOUT pin is logic 1

Table 23-21. SPIPC4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ENASET	R/W	0h	SPIENA dataout set. Only active when the SPIENA pin is configured as a general-purpose output pin. A value of '1' written to this bit sets the corresponding ENABLEDOUT bit to '1'. Write: 0= Has no effect 1= Logic 1 placed on SPIENA pin if it is in General Purpose O/P mode Read: 0= Current value on ENADOUT is 0. 1= Current value on ENADOUT is 1
7-0	SCSSET	R/W	0h	SPISCS[7:0] dataout set. Only active when the SPISCSx pins are configured as a general-purpose output pins. A value of '1' written to these bits set the corresponding SCSDOUT bit to '1'. Write: 0= Has no effect 1= Logic 1 placed on SPISCSx pin if it is in General Purpose O/P mode Read: 0= Current value on SCSDOUTx is 0. 1= Current value on SCSDOUTx is 1. Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSSET[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always. Note: Register Read Read of SPIPC4 register gives out contents of the SPIPC3 register.

23.3.1.11 SPIPC5 Register (Offset = 28h) [reset = 0h]

SPIPC5 is shown in Figure 23-108 and described in Table 23-90.

Return to Summary Table.

SPI / MibSPI Pin Control Register 5 (SPIPC5) - SPIDCLR

Figure 23-42. SPIPC5 Register

31	30	29	28	27	26	25	24
SOMICLR							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOCLR							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMICLR0	SIMOCLR0	CLKCLR	ENACLR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSCLR							
R/W-0h							

Table 23-22. SPIPC5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMICLR	R/W	0h	SPISOMIx dataout clear. Only active when the SPISOMIx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SPISOMIDOUTx bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISOMIx pin if it is in General Purpose O/P mode Read: 0= Current value on SOMIDOUTx is 0. 1= Current value on SOMIDOUTx is 1.
23-16	SIMOCLR	R/W	0h	SPISIMOx dataout clear. Only active when the SPISIMOx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SPISIMODOUTx bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISIMOx pin if it is in General Purpose O/P mode Read: 0=Current value on SIMODOUTx is 0. 1=Current value on SIMODOUTx is 1.
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMICLR0	R/W	0h	SPISOMI0 dataout clear. Only active when the SPISOMI0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SPISOMIDOUT bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISOMI0 pin if it is in General Purpose O/P mode Read: 0= Current value on SOMIDOUT0 is 0. 1= Current value on SOMIDOUT0 is 1. Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOCLR0	R/W	0h	SPISIMO0 dataout clear. Only active when the SPISIMO0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SIMODOUT0 bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISIMO0 pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUT0 is 0. 1= Current value on SIMODOUT0 is 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKCLR	R/W	0h	SPICLK dataout clear. Only active when the SPICLK pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding CLKDOUT bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPICLK pin if it is in General Purpose O/P mode Read: 0= Current value on CLKDOUT is 0. 1= Current value on CLKDOUT is 1.

Table 23-22. SPIPC5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ENACLR	R/W	0h	SPIENA dataout clear. Only active when the SPIENA pin is configured as a general-purpose output pin. A value of '1' written to this bit clears the corresponding ENABLEDOUT bit to '0'. Write: 0= Has no effect 1= Logic 0 placed on SPIENA pin if it is in General Purpose O/P mode Read: 0= Current value on ENADOUT is 0. 1= Current value on ENADOUT is 1.
7-0	SCSCLR	R/W	0h	SPISCS[7:0] dataout clear. Only active when the SPISCSx pins are configured as a general-purpose output pins. A value of '1' written to this bit clears the corresponding SCSDOUT bit to '0.. Write: 0= Has no effect 1= Logic 0 placed on SPISCSx pin if it is in General Purpose O/P mode Read: 0= Current value on SCSDOUTx is 0. 1= Current value on SCSDOUTx is 1 Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSCLR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always. Note: Register Read Read of SPIPC5 register gives out contents of the SPIPC3 register.

23.3.1.12 SPIPC6 Register (Offset = 2Ch) [reset = 0h]

SPIPC6 is shown in Figure 23-109 and described in Table 23-91.

Return to Summary Table.

SPI / MibSPI Pin Control Register 6 (SPIPC6) - SPIPDR

Figure 23-43. SPIPC6 Register

31	30	29	28	27	26	25	24
SOMIPDR							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOPDR							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIPDR0	SIMOPDR0	CLKPDR	ENAPDR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSPDR							
R/W-0h							

Table 23-23. SPIPC6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIPDR	R/W	0h	SPISOMIx Open drain enable Enables Open drain capability for the pin SOMIx if the following conditions are met. SOMIDIRx = 1 (SPISOMI0 pin configured in GPIO mode as output pin) SOMIDOUTx = 1 0 = Output value on SPISOMIx pin is logic ‘1’ 1 = Output pin SPISOMIx is Tri-stated
23-16	SIMOPDR	R/W	0h	SPISIMOx Open drain enable Enables Open drain capability for the pin SPISIMOx if the following conditions are met. SIMODIRx = 1 (SPISIMOx pin configured in GPIO mode as output pin) SIMODOUTx = 1 0 = Output value on SPISIMOx pin is logic ‘1’ 1 = Output pin SPISIMOx is Tri-stated
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIPDR0	R/W	0h	SPISOMI0 Open drain enable Enables Open drain capability for the pin SPISOMI if the following conditions are met. SOMIDIR0 = 1 (SPISOMI0 pin configured in GPIO mode as output pin) SOMIDOUT0 = 1 0 = Output value on SPISOMI0 pin is logic ‘1’ 1 = Output pin SPISOMI0 is Tri-stated Note: Bit 11 or bit 24 can be used to set the direction for pin SPISOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOPDR0	R/W	0h	SPISIMO0 Open drain enable Enables Open drain capability for the pin SPISIMO0 if the following conditions are met. SIMODIR0 = 1 (SPISIMO pin configured in GPIO mode as output pin) SIMODOUT0 = 1 0 = Output value on SPISIMO0 pin is logic ‘1’ 1 = Output pin SPISIMO0 is Tri-stated Note: Bit 10 or bit 16 can be used to set the direction for pin SPISIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKPDR	R/W	0h	SPICLK Open drain enable Enables Open drain capability for the pin CLK if the following conditions are met. CLKDIR = 1 (SPICLK pin configured in GPIO mode as output pin) CLKDOUT = 1 0 = Output value on SPICLK pin is logic ‘1’ 1 = Output pin SPICLK is Tri-stated
8	ENAPDR	R/W	0h	SPIENA Open drain enable Enables Open drain capability for the pin SPIENA if the following conditions are met. ENABLEDIR = 1 (SPIENA pin configured in GPIO mode as output pin) ENABLEDOUT = 1 0 = Output value on SPIENA pin is logic ‘1’ 1 = Output pin SPIENA is Tri-stated

Table 23-23. SPIPC6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	SCSPDR	R/W	0h	SPISCSx Open drain enable Enables Open drain capability for the pin SPISCSx if the following conditions are met. SCSDIRx = 1 (SPISCS pin configured in GPIO mode as output pin) SCSDOUTx = 1 0 = Output value on SPISCSx pin is logic '1' 1 = Output pin SPISCSx is Tri-stated Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSPDR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always.

23.3.1.13 SPIDAT0 Register (Offset = 38h) [reset = 0h]

SPIDAT0 is shown in Figure 23-110 and described in Table 23-92.

Return to Summary Table.

SPI / MibSPI Transmit Data Register 0 Note: Accessibility of SPIDAT0 The SPIDAT0 register is not accessible in Multibuffer Mode of MibSPI. It is only accessible in compatibility mode.

Figure 23-44. SPIDAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																TXDATA															
R-0h																R/W-0h															

Table 23-24. SPIDAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
15-0	TXDATA	R/W	0h	SPI / MibSPI Transmit Data. When written, these bits will be copied to the Shift Register if it is empty. If the Shift Register is not empty, the TXBUF will hold the written values. SPIEN (SPICGR1.24) must be set to 1 before this register can be written to. Writing a 0 to the SPIEN register forces the lower 16 bits of the SPIDAT0 register to 0x00. When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned. As the data is shifted out from either the MSB or the LSB of Transmit Shift Register depending upon SHIFTDIR bit (SPIFMTx.20). Simultaneously, the received bit will be shifted into the Receive Shift Register either through LSB or MSB depending upon SHIFTDIR bit. This allows the concurrent transmission and reception of data. Note: Irrespective of the character length, the Transmit data should be right justified before writing to SPIDAT0 register. The default Data Format Control register for SPIDAT0 is SPIFMT0. However it is possible to reprogram the DFSEL[1:0] fields of SPIDAT1 prior to using SPIDAT0, to select a different SPIFMTx register.

23.3.1.14 SPIDAT1 Register (Offset = 3Ch) [reset = 0h]

SPIDAT1 is shown in Figure 23-111 and described in Table 23-93.

Return to Summary Table.

SPI / MibSPI Transmit Data Register 1 When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned.

Figure 23-45. SPIDAT1 Register

31	30	29	28	27	26	25	24
NU2			CSHOLD	NU1	WDEL	DFSEL	
R-0h			R/W-0h	R-0h	R/W-0h	R/W-0h	
23	22	21	20	19	18	17	16
CSNR							
R/W-0h							
15	14	13	12	11	10	9	8
TXDATA							
R/W-0h							
7	6	5	4	3	2	1	0
TXDATA							
R/W-0h							

Table 23-25. SPIDAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
28	CSHOLD	R/W	0h	Chip select hold mode. In SPI or compatibility mode MibSPI, the CSHOLD bit is supported in master mode only. In slave mode, this bit is ignored. CSHOLD defines the behavior of the chip select line at the end of a data transfer. 1 =The chip select signal is held active at the end of a transfer until a control field with new data and control information is loaded into SPIDAT1. If the new chip select information equals the previous one, the active chip select signal is extended until the end of transfer with CSHOLD cleared or until the chip select information changes. 0 =The chip select signal is deactivated at the end of a transfer after the T2CDELAY time has passed. If two consecutive transfers are dedicated to the same chip select this chip select signal will be deactivated for atleast 2VCLK cycles before it is activated again.
27	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
26	WDEL	R/W	0h	Enable the delay counter at the end of the current transaction. WDELAY bit is supported in Master mode only. In Slave mode, this bit will be ignored. 1 = After a transaction, WDELAY of the corresponding data format will be loaded into the delay counter. No transaction will be performed until the WDELAY counter overflows. The SPISCS pins will be de-activated for atleast (WDELAY + 2) * VCLK_Period duration. 0 = No delay will be inserted. However, SPISCS pins will still be de-activated for atleast for 2VCLK cycles if CSHOLD = ‘0’. In SPI or Compatibility mode of MibSPI, the duration for which the SPISCS pin remaining de-activated will also depend upon time taken to supply a new data after completing the shifting operation. If the internal buffer - TXBUF is already full, then the SPISCS will be deasserted for atleast 2 VCLK cycles(if WDEL = ‘0’).
25-24	DFSEL	R/W	0h	DFSEL1 DFSEL0 Description 0 0 Data word format 0 is selected 0 1 Data word format 1 is selected 1 0 Data word format 2 is selected 1 1 Data word format 3 is selected

Table 23-25. SPIDAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-16	CSNR	R/W	0h	Chip select number. CSNR defines the chip select that shall be activated during the data transfer. The value of CSNR[7:0] will be driven on SPISCS[7:0] lines during the transfer. Note: Effect of NUM_CS_PINS generic on CSNR bits. Actual number of bits implemented in CSNR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always. Note: Preselecting a Format Register. Writing to just the Control Field (using byte writes) does not initiate any SPI transfer in Master mode. This feature can be used to setup SPICLK Phase or Polarity before actually starting the transfer by just updating the DFSEL fields in the control field to select the required Phase/Polarity combination.
15-0	TXDATA	R/W	0h	SPI / MibSPI Transmit Data. When written, these bits will be copied to the Shift Register if it is empty. If the Shift Register is not empty, the TXBUF will hold the written values. SPIEN must be set to 1 before this register can be written to. Writing a 0 to the SPIEN register forces the lower 16 bits of the SPIDAT1 register to 0x00. Write to this register ONLY when using the automatic Slave Chip Select feature. See section 3, Operation Modes, on page 14. A write to this register will drive the contents of CSNR[7:0] into the respective pins in SPISCS[7:0] if those are configured as functional pins. When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned. Irrespective of the character length, the Transmit data should be right justified before writing to SPIDAT1 register.

23.3.1.15 SPIBUF Register (Offset = 40h) [reset = 80000000h]

SPIBUF is shown in Figure 23-112 and described in Table 23-94.

Return to Summary Table.

SPI / MibSPI Receive Buffer Register

Figure 23-46. SPIBUF Register

31	30	29	28	27	26	25	24
RXEMPTY	RXOVR	TXFULL	BITERR	DESYNC	PARITYERR	TIMEOUT	DLENERR
1h	0h	R-0h	0h	0h	0h	0h	0h
23	22	21	20	19	18	17	16
LCSNR							
R-0h							
15	14	13	12	11	10	9	8
RXDATA							
R-0h							
7	6	5	4	3	2	1	0
RXDATA							
R-0h							

Table 23-26. SPIBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RXEMPTY		1h	Receive data buffer empty. When host reads the SPIBUF field or the whole SPIBUF register this will automatically set the RXEMPTY flag. When a data transfer is completed, the received data is copied into SPIBUF, the RXEMPTY flag is cleared. 1 =No data received since last reading of SPIBUF register. 0 =A new Data is received and copied into SPIBUF field. This flag gets set to '1' under following conditions: oReading RXDATA portion of the SPIBUF register. oWriting '1' to clear the RXINTFLG bit in SPIFLG register. Write-Clearing the RXINTFLG bit before reading the SPIBUF indicates the received data is being ignored. Conversely, RXINTFLG can be cleared by reading the RXDATA portion of the SPIBUF register. So, reading the full of SPIBUF register itself clears the Receiver Full Interrupt Flag.
30	RXOVR		0h	Receive data buffer overrun. When a data transfer is completed and the received data is copied into the RXBUF while it is already full, RXOVR is set. Refer to Figure 1 for a view of internal logic diagram. An Overrun always occurs to the RXBUF, and SPIBUF contents never get Overwritten until after it is read by the VBUSP Master. If enabled, RXOVRN interrupt gets generated when RXBUF is Overwritten, and reading SPIFLG or SPIVECTx register shows the RXOVRN condition. However, two read operations to the SPIBUF register are required to reach the Overrun buffer. 1 =A receive data overrun condition occurred since last time reading the data field. 0 =No receive data overrun condition occurred since last time reading the data field. This bit is cleared to '0' under the following conditions: o RXDATA portion of the SPIBUF register is read. o OVRNINTFLG bit in SPIFLG register is write-cleared. When an Overrun occurs, the SPIBUF contents will not be overwritten. Only the RXBUF contents are overwritten. When the SPIBUF is read out , the RXBUF contents get copied to the SPIBUF if RXBUF is full. So, the first data read out will be intact when an Overrun occurs. Note: A special condition under which RXOVR flag gets set. If both SPIBUF & RXBUF are already full and while another buffer receive is underway, if any errors like TIMEOUT, BITERR & DLEN_ERR occur, then RXOVR in RXBUF & SPIFLG registers will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a normal Receiver Overrun.

Table 23-26. SPIBUF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	TXFULL	R	0h	Transmit data buffer full. This flag is a read-only flag. Writing into SPIDAT0 or SPIDAT1 field while TX Shift Register is full will automatically set the TXFULL flag. Once the data is copied to the Shift Register the TXFULL flag will be cleared. Writing to SPIDAT0/SPIDAT1 register when both TXBUF & the TX Shift Register are empty does not set the TXFULL flag. 1 =Transmit buffer is full, SPIDAT0/SPIDAT1 is not ready to accept a new data. 0 =Transmit buffer is empty, SPIDAT0/SPIDAT1 is ready to accept a new data.
28	BITERR		0h	Mismatch of internal transmit data and transmitted data. 1 =A bit error occurred. The SPI / MibSPI samples the signal of the transmit pin (master: SIMO, slave: SOMI) at the receive point (half clock cycle after transmit point). If the sampled value differs from the transmitted value a bit error is detected and the flag BITERR is set. A possible reason for a bit error can be noise, a to high bit rate / capacitive load or another master/slave trying to transmit at the same time. 0 =No bit error occurred. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read.
27	DESYNC		0h	De-synchronization of slave device. De-synchronization monitor is active in master mode only. 1 =A slave device is de-synchronized. The master monitors the ENA signal coming from the slave device and sets the DESYNC flag if ENA is deactivated before the last reception point or after the last bit is transmitted plus tT2EDELAY (see Section 8.21). If DESYNCENA is set an interrupt is asserted. De-synchronization can occur if a slave device misses a clock edge coming from the master. 0 =No slave de-synchronization detected. This bit is valid in Master mode only. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read. Note: Possible inconsistency of Desync flag in SPI/Compatibility mode MibSPI Due to the nature of this Error, under some circumstances it is possible for Desync error detected for the previous buffer to be visible in the current buffer. This is due to the fact that Receive Completion flag/interrupt will be generated when the buffer transfer is completed. But Desync will be detected after the buffer transfer is completed. So, if VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct Desync condition. This inconsistency in Desync flag is valid only in SPI or Compatibility mode of MibSPI. In Multibuffer mode, Desync flag is always guaranteed to be for the current buffer.
26	PARITYERR		0h	Calculated parity differs from received parity bit. 1 =A parity error occurred. If the parity generator is enabled (can be selected individually for each buffer) an even or odd parity bit is added at the end of a data word (see Section 8.23). During reception of the data word the parity generator calculates the reference parity and compares it to the received parity bit. In the event of a mismatch the PARITYERR flag is set. 0 =No parity error detected. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read.
25	TIMEOUT		0h	Time-out due to non-activation of ENA pin. 1 =An ENA signal time-out occurred. The SPI / MibSPI generates a time-out because the slave hasn’t responded in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition the TIMOUT flag in the status field of the corresponding buffer and in the SPIFLG register is set. 0 =No ENA-pin time-out occurred. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read. This bit is valid in Master mode only.
24	DLENERR		0h	Data Length Error flag. 1 = A Data Length Error has occured. 0 = No Data Length Error has occured. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read.

Table 23-26. SPIBUF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-16	LCSNR	R	0h	Last Chip select number. LCSNR in the status field is a copy of CSNR in the corresponding control field. It defines the chip select that has been activated during the last data transfer from the corresponding buffer. This is the copy of CSNR bits from SPIDAT1 latched at the end of a transfer. Note: Effect of NUM_CS_PINS generic on LCSNR bits. Actual number of bits implemented in LCSNR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will read '0' always
15-0	RXDATA	R	0h	SPI Receive Data. This is the received data, transferred from the Receive Shift-Register at the end of a transfer completion. Irrespective of the programmed character length & the direction of shifting, the received data is stored right-justified in the register.

23.3.1.16 SPIEMU Register (Offset = 44h) [reset = 80000000h]

SPIEMU is shown in Figure 23-113 and described in Table 23-95.

Return to Summary Table.

SPI / MibSPI Emulation Register Note: All the fields ot SPIEMU register are Read-Only. Read operation on this register under any mode will not have any impact on the status of this or any other registers.

Figure 23-47. SPIEMU Register

31	30	29	28	27	26	25	24
RXEMPTY	RXOVR	TXFULL	BITERR	DESYNC	PARITYERR	TIMEOUT	DLENERR
R-1h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
LCSNR							
R-0h							
15	14	13	12	11	10	9	8
RXDATA							
R-0h							
7	6	5	4	3	2	1	0
RXDATA							
R-0h							

Table 23-27. SPIEMU Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RXEMPTY	R	1h	Receive data buffer empty. 1 = No data received since last reading of SPIBUF register. 0 = A new Data is received and copied into SPIBUF field.
30	RXOVR	R	0h	Receive data buffer overrun. 1 =A receive data overrun condition occurred since last time reading the data field. 0 =No receive data overrun condition occurred since last time reading the data field.
29	TXFULL	R	0h	Transmit data buffer full. 1 =Transmit buffer is full, SPIDAT0/SPIDAT1 is not ready to accept a new data. 0 =Transmit buffer is empty, SPIDAT0/SPIDAT1 is ready to accept a new data.
28	BITERR	R	0h	Mismatch of internal transmit data and transmitted data. 1 =A bit error occurred. The SPI / MibSPI samples the signal of the transmit pin (master: SIMO, slave: SOMI) at the receive point (half clock cycle after transmit point). If the sampled value differs from the transmitted value a bit error is detected and the flag BITERR is set. A possible reason for a bit error can be noise, a to high bit rate / capacitive load or another master/slave trying to transmit at the same time. 0 =No bit error occurred.
27	DESYNC	R	0h	De-synchronization of slave device. De-synchronization monitor is active in master mode only. 1 =A slave device is de-synchronized. The master monitors the ENA signal coming from the slave device and sets the DESYNC flag if ENA is deactivated before the last reception point or after the last bit is transmitted plus tT2EDELAY (see Section 8.21). If DESYNCENA is set an interrupt is asserted. De-synchronization can occur if a slave device misses a clock edge coming from the master. 0 =No slave de-synchronization detected. This bit is valid in Master mode only.
26	PARITYERR	R	0h	Calculated parity differs from received parity bit. 1 =A parity error occurred. If the parity generator is enabled (can be selected individually for each buffer) an even or odd parity bit is added at the end of a data word (see Section 8.23). During reception of the data word the parity generator calculates the reference parity and compares it to the received parity bit. In the event of a mismatch the PARITYERR flag is set. 0 =No parity error detected.

Table 23-27. SPIEMU Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	TIMEOUT	R	0h	Time-out due to non-activation of ENA pin. 1 =An ENA signal time-out occurred. The SPI / MibSPI generates a time-out because the slave hasn't responded in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition the TIMOUT flag in the status field of the corresponding buffer and in the SPIFLG register is set. 0 =No ENA-pin time-out occurred. This bit is valid in Master mode only.
24	DLENERR	R	0h	Data Length Error flag. 1 = A Data Length Error has occured. 0 = No Data Length Error has occured.
23-16	LCSNR	R	0h	Last Chip select number. LCSNR in the status field is a copy of CSNR in the corresponding control field. It defines the chip select that has been activated during the last data transfer from the corresponding buffer. This is the copy of CSNR bits from SPIDAT1 latched at the end of a transfer. Note: Effect of NUM_CS_PINS generic on LCSNR bits. Actual number of bits implemented in LCSNR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will read '0' always.
15-0	RXDATA	R	0h	SPI Receive Data. SPI / MibSPI emulation is a mirror of the SPIBUF register. The only difference between SPIEMU and SPIBUF is that a read from SPIEMU does not clear any of the status flags.

23.3.1.17 SPIDELAY Register (Offset = 48h) [reset = 0h]

SPIDELAY is shown in Figure 23-114 and described in Table 23-96.

Return to Summary Table.

SPI / MibSPI Delay Register

Figure 23-48. SPIDELAY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2TDELAY								T2CDELAY								T2EDELAY								C2EDELAY							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 23-28. SPIDELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	C2TDELAY	R/W	0h	Chip-select-active-to-transmit-start-delay. C2TDELAY is used in master mode only. It defines a setup time for the slave device that delays the data transmission from the chip select active edge by a multiple of VBUSPCLK cycles. ChipSelect-active-to-transmission delays between 2 to 257 VBUSPCLK cycles can be achieved. If Phase = '1', the delay between SCS fall-edge to the first edge of SPICLK will have an additional 0.5 SPICLK Period delay. This delay is as per the SPI protocol.
23-16	T2CDELAY	R/W	0h	Transmit-end-to-chip-select-inactive-delay. T2CDELAY is used in master mode only. It defines a hold time for the slave device that delays the chip select deactivation by a multiple of VBUSPCLK cycles after the last bit is transferred. T2CDELAY can be configured between 2 and 256 VBUSPCLK cycles. If Phase = '0', then between the last edge of SPICLK and rise-edge of SCS, there will be an additional delay of 0.5 SPICLK period. This is as per the SPI protocol Both C2TDELAY and T2CDELAY counters do not have any dependancy on SPIENA pin value. Even if the SPIENA pin is asserted by the Slave, Master will continue to delay the start of SPICLK until the C2TDELAY counter overflows. Similarly, even if the SPIENA pin is deasserted by the Slave, Master will continue to hold the SPISCS pins active until the T2CDELAY counter overflows. This way, it is guaranteed that the setup/hold times of the SPISCS pins is determined by the Delay timers alone. To achieve better throughput, it should be ensured that these two timers are kept at the minimum possible values.
15-8	T2EDELAY	R/W	0h	Transmit-data-finished-to-ENA-pin-inactive-time-out. T2EDELAY is used in master mode only. It defines a time-out value as a multiple of SPI clock before the ENAble signal has to become inactive and after the CS becomes inactive. The SPI clock depends on which data format is selected. If the slave device is missing one or more clock edges, it is becoming de-synchronized. Although the master has finished the data transfer the slave is still waiting for the missed clock pulses and the ENA signal isn't disabled. The T2EDELAY defines a time-out value that triggers the DESYNC flag, if the ENA signal isn't deactivated in time. DESYNC flag is set to indicate that the Slave device did not deassert its SPIENA pin in time to acknowledge that it has received all the bits of the sent character. If T2CDELAY is programmed a non-zero value, then T2EDELAY will start only after the T2CDELAY completes. This should be taken into consideration to determine an optimum value of T2EDELAY. Note: Zero T2EDELAY If T2EDELAY is not programmed or programmed to '0', then the Master SPI/MibSPI does not wait for SPIENA pin to be deasserted. It ignores the state of the SPIENA pin after the transmission is completed.

Table 23-28. SPIDELAY Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	C2EDELAY	R/W	0h	Chip-select-active-to-ENA-signal-active-time-out C2EDELAY is utilized only in master mode and it applies only if the addressed slave generates an ENA signal as a hardware handshake response. C2EDELAY defines the maximum time between the SPI / MibSPI activating the chip select signal and the addressed slave responding by activating the ENA signal. C2EDELAY defines a time-out value as a multiple of SPI clocks. The SPI clock depends on whether data format 0 or data format 1 is selected. If the slave device is not responding with the ENA signal before the time-out value is reached, the TIMEOUT flag in SPIFLG register is set and an interrupt is asserted if enabled. If a time-out occurs the MibSPI clears the transmit request of the timed-out buffer, sets the TIMEOUT flag for the current buffer and continues with the transfer of the next buffer in the sequence that is enabled. If C2TDELAY is programmed a non-zero value, then C2EDELAY will start only after the C2TDELAY completes. This should be taken into consideration to determine an optimum value of C2EDELAY. Note: Zero C2EDELAY If C2EDELAY is not programmed or programmed to ‘0’, then the Master SPI/MibSPI waits until the SPIENA pin to asserted before it can start the transfer. If the SPIENA pin is not asserted due to a malfunctioning Slave, the Master will wait forever. This might cause hang-up situation. it is recommended to program to C2EDELAY to a suitable value whenever SPIENA pin is used as functional.

23.3.1.18 SPIDEF Register (Offset = 4Ch) [reset = 1h]

SPIDEF is shown in Figure 23-115 and described in Table 23-97.

Return to Summary Table.

SPI / MibSPI Default Chip select Register

Figure 23-49. SPIDEF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								CSDEF0							
R-0h																								R/W-1h							

Table 23-29. SPIDEF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
7-0	CSDEF0	R/W	1h	Chip select default pattern. Master mode behavior. The CSDEFx bits are output to the chip select pins when no transmission are currently performed. It allows the user to set a chip select pattern which deselects all the SPI slaves. 1 =If CSDEFx is set to “1” the corresponding chip select is set to “1” while SPI/MibSPI is IDLE. 0 =If CSDEFx is set to “0” the corresponding chip select is set to “0” while SPI/MibSPI is IDLE. Note: Effect of NUM_CS_PINS generic on CSDEF bits. Actual number of bits implemented in CSDEF[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be treated as reserved, read-only and will read ‘0’ always.

23.3.1.19 SPIFMT0 Register (Offset = 50h) [reset = 0h]

SPIFMT0 is shown in Figure 23-116 and described in Table 23-98.

Return to Summary Table.

SPI / MibSPI Data Format Register 0

Figure 23-50. SPIFMT0 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-30. SPIFMT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-30. SPIFMT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.1.20 SPIFMT1 Register (Offset = 54h) [reset = 0h]

SPIFMT1 is shown in Figure 23-117 and described in Table 23-99.

Return to Summary Table.

SPI / MibSPI Data Format Register 1

Figure 23-51. SPIFMT1 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-31. SPIFMT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-31. SPIFMT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.1.21 SPIFMT2 Register (Offset = 58h) [reset = 0h]

SPIFMT2 is shown in Figure 23-118 and described in Table 23-100.

Return to Summary Table.

SPI / MibSPI Data Format Register 2

Figure 23-52. SPIFMT2 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-32. SPIFMT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-32. SPIFMT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.1.22 SPIFMT3 Register (Offset = 5Ch) [reset = 0h]

SPIFMT3 is shown in Figure 23-119 and described in Table 23-101.

Return to Summary Table.

SPI / MibSPI Data Format Register 3

Figure 23-53. SPIFMT3 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-33. SPIFMT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-33. SPIFMT3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.1.23 TGINTVECT0 Register (Offset = 60h) [reset = 0h]

TGINTVECT0 is shown in Figure 23-120 and described in Table 23-102.

Return to Summary Table.

SPI Interrupt Vector Register 0 / MibSPI Transfer Group Interrupt Vector Register 0

Figure 23-54. TGINTVECT0 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU				INTVECT0			SUSPEND0
R-0h				R-0h			R-0h

Table 23-34. TGINTVECT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
5-1	INTVECT0	R	0h	Interrupt vector for interrupt line INT0. INTVECT0 returns the vector of the pending interrupt at interrupt line INT0. If more than one interrupts are pending, INTVECT0 always references the highest priority interrupt source first. The vectors generated are different for SPI / Compatibility mode of MibSPI and Multibuffer mode of MibSPI. INTVECT0 field just reflects the status of SPIFLG register in a vectorized format. So, any updates to SPIFLG will automatically reflect in the Vector value in this register. SPI / Compatibility mode - MibSPI The interrupts available for SPI or Compatibility mode - MibSPI, in the descending order of their priorities are as given below. Transmission Error Interrupt Receive Buffer Overrun Interrupt Receive Buffer Full Interrupt Transmit Buffer Empty Interrupt Vectors for each of these interrupts will be reflected on the INTVECT0 bits, when they occur. Reading the vectors for the “Receive Buffer Overrun” & “Receive Buffer Full” interrupts will automatically clear the respective flags in the SPIFLG register. On reading the INTVECT0 bits, the vector of the next highest priority interrupt (if any) will be then reflected on the INTVECT0 bits. If two or more interrupts occur simultaneously, the vector for the higest priority interrupt will be reflected on the INTVECT0 bits. Reading the Vector register when “Transmitter Empty” is indicated does not clear the TXINTFLG in SPIFLG register. Writing a new data to SPIDATx register clears the “Transmitter Empty” interrupt. INTVECT0[4:0] Description 00000b no interrupt pending 10001b Error interrupt pending. Refer to Least Significant (LS) Byte of SPIFLG to determine more details about the type of error. 10011b Pending interrupt is “Receive Buffer Overrun Interrupt” 10010b Pending interrupt is “Receive Buffer Full Interrupt” 10100b Pending interrupt is “Transmit Buffer Empty Interrupt” All other bit combinations Reserved Note: Exception for clearing of RXINT If both SPIBUF and RXBUF (internal buffer) are full, then, reading TGINTVECT0 register (while it shows 10010) does not clear the RXINTFLG in SPIFLG register. In this case, only way to clear the Interrupt is to read the SPIBUF again until there’s no more unread RX data.

Table 23-34. TGINTVECT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SUSPEND0	R	0h	“Transfer suspended” or “transfer finished” interrupt.(MibSPI only) The SUSPEND0 flag is updated depending on the type of interrupt reflected by the VECTOR value field. 1 =The interrupt type is a “transfer suspended” interrupt. I.e. the transfer group referenced by INTVECT0 has asserted an interrupt, because the buffer to be transferred next is in “suspend to wait” mode. 0 =The interrupt type is a “transfer finished” interrupt. I.e. the buffer array referenced by INTVECT0 has asserted an interrupt, because all of the data from the whole transfer group has been transferred. Note: Special case for SUSPEND interrupt When there’s a “transfer suspended” interrupt (SUSPEND0 bit is set to ‘1’), reading the TGINTVECT0 register does not clear the “TG Suspended” interrupt. The SUSPEND condition should be resolved first before the interrupt can be cleared. The SUSPEND condition can be cleared by writing a new data to that TXRAM location and/or reading the data from that RXRAM location depending upon the SUSPEND criteria programmed in the “BUFMODE” field of that TXRAM location. The SUSPEND0 bit always returns value ‘0’ in SPI / Compatibility mode MibSPI. When there is an RXOVRN or any Error interrupt in Multibuffer mode, SUSPEND0 bit stays ‘0’. Refer to Section 6.2 on page 77 & Section 6.3 on page 77 for more details and notes on better handling of interrupts. Note: Reading Error Vector Reading an Error Vector in the TGINTVECT0 register will NOT clear the Error flags in the SPIFLG register. The Error Flags in SPIFLG need to be write-cleared after servicing them suitably. If “TG Completed” interrupt occurs for a TG and after a while “TG Suspended” flag too gets set for the same TG, then the TGINTVECT0 register will show “TG Completed” interrupt giving it higher priority than the “TG Suspended” interrrupt.

23.3.1.24 TGINTVECT1 Register (Offset = 64h) [reset = 0h]

TGINTVECT1 is shown in Figure 23-121 and described in Table 23-103.

Return to Summary Table.

SPI Interrupt Vector Register 1 / MibSPI Transfer Group Interrupt Vector Register 1

Figure 23-55. TGINTVECT1 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU		INTVECT1					SUSPEND1
R-0h		R-0h					R-0h

Table 23-35. TGINTVECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
5-1	INTVECT1	R	0h	Interrupt vector for interrupt line INT1. INTVECT1 returns the vector of the pending interrupt at interrupt line INT0. If more than one interrupt is pending, INTVECT1 always references the highest priority interrupt source first. The vectors generated are different for SPI / Compatibility mode of MibSPI and Multibuffer mode of MibSPI. INTVECT1 field just reflects the status of SPIFLG register in a vectorized format. So, any updates to SPIFLG will automatically reflect in the Vector value in this register. SPI / Compatibility mode - MibSPI The interrupts available for SPI / Compatibility mode - MibSPI, in the descending order of their priorities are as given below. Transmission Error Interrupt Receive Buffer Overrun Interrupt Receive Buffer Full Interrupt Transmit Buffer Empty Interrupt Vectors for each of these interrupts will be reflected on the INTVECT1 bits, when they occur. Reading the vectors for the “Receive Buffer Overrun” & “Receive Buffer Full” interrupts will automatically clear the respective flags in the SPIFLG register. On reading the INTVECT1 bits, the vector of the next highest priority interrupt (if any) will then be reflected on the INTVECT1 bits. If two or more interrupts occur simultaneously, the vector for the higest priority interrupt will be reflected on the INTVECT1 bits. Reading the Vector register when “Transmitter Empty” is indicated does not clear the TXINTFLG in SPIFLG register. Writing a new data to SPIDATx register clears the “Transmitter Empty” interrupt. 00000b no interrupt pending 10001b Error interrupt pending. Refer to LS Byte of SPIFLG to determine more details about the type of error. 10011b Pending interrupt is “Receive Buffer Overrun Interrupt” 10010b Pending interrupt is “Receive Buiffer Full Interrupt” 10100b Pending interrupt is “Transmit Buffer Empty Interrupt” All other bit combinations Reserved Note: Exception for clearing of RXINT If both SPIBUF and RXBUF (internal buffer) are full, then, reading TGINTVECT1 register (while it shows 10010) does not clear the RXINTFLG in SPIFLG register. In this case, only way to clear the Interrupt is to read out the SPIBUF again until there’s no more unread RX data.

Table 23-35. TGINTVECT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SUSPEND1	R	0h	“Transfer suspended” or “transfer finished” interrupt.(MibSPI Only) The SUSPEND1 flag is updated depending on the type of interrupt reflected by the VECTOR value field. 1 =The interrupt type is a “transfer suspended” interrupt. I.e. the transfer group referenced by INTVECT1 has asserted an interrupt, because the buffer to be transferred next is in “suspend to wait” mode. 0 =The interrupt type is a “transfer finished” interrupt. I.e. the buffer array referenced by INTVECT1 has asserted an interrupt, because all data from the whole transfer group has been transferred. Note: Special case for SUSPEND interrupt When there’s a “transfer suspended” interrupt(SUSPEND1 bit is set to ‘1’), reading the TGINTVECT1 register does not clear the “TG Suspended” interrupt. The SUSPEND condition should be resolved first before the interrupt can be cleared. This condition can be cleared by writing a new data to that TXRAM location and/or reading the data from that RXRAM location depending upon the SUSPEND criteria programmed in the “BUFMODE” field of that TXRAM location. The SUSPEND1 bit always returns value ‘0’ in SPI / Compatibility mode MibSPI. Even while there is an RXOVRN or any Error interrupt in Multibuffer mode, SUSPEND1 bit stays ‘0’. Refer to Section 6.2 on page 77 & Section 6.3 on page 77 for more details and notes on better handling of interrupts. Note: Reading Error Vector Reading an Error Vector in the TGINTVECT1 register will NOT clear the Error flags in the SPIFLG register. The Error Flags in SPIFLG need to be write-cleared after servicing them suitably. If “TG Completed” interrupt occurs for a TG and after a while “TG Suspended” flag too gets set for the same TG, then the TGINTVECT1 register will show “TG Completed” interrupt giving it higher priority than the “TG Suspended” interrrupt.

23.3.1.25 SPIPC9 Register (Offset = 68h) [reset = 0h]

SPIPC9 is shown in Figure 23-122 and described in Table 23-104.

Return to Summary Table.

SPI/MibSPI Pin Control Register 9 (SPIPC9) - SPISRSEL

Figure 23-56. SPIPC9 Register

31	30	29	28	27	26	25	24
SOMISRS7							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOSRS7							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMISRS0	SIMOSRS0	CLKSRS	ENASRS
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSSRS							
R/W-0h							

Table 23-36. SPIPC9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMISRS7	R/W	0h	Each of these 7 bits controls the slew rate for the corresponding SPISOMIx pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select.
23-16	SIMOSRS7	R/W	0h	Each of these 7 bits controls the slew rate for the corresponding SPISIMOx pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select.
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMISRS0	R/W	0h	This bit controls the slew rate for SPISOMI0 pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select. Note: Bit 11 or bit 24 can be used to control the slew rate for SPISOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOSRS0	R/W	0h	This bit controls the slew rate for SPISIMO0 pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select. Note: Bit 10 or bit 16 can be used to control the slew rate for SPISIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKSRS	R/W	0h	This bit controls the slew rate for SPICLK pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select.
8	ENASRS	R/W	0h	This bit controls the slew rate for SPIENA pin. 0 =Fast Buffer Select. 1 =Slow Buffer Select.
7-0	SCSSRS	R/W	0h	Each of these 7 bits controls the slew rate for the corresponding SPISCSx pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select. Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSSRS[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read ‘0’ always.

23.3.1.26 SPIPMCTRL Register (Offset = 6Ch) [reset = 0h]

SPIPMCTRL is shown in Figure 23-123 and described in Table 23-105.

Return to Summary Table.

SPI/MibSPI Parallel/Modulo Mode Control Register

Figure 23-57. SPIPMCTRL Register

31	30	29	28	27	26	25	24
NU4	HSM_MODE3	MODCLKPOL3	MMODE3			PMODE3	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	
23	22	21	20	19	18	17	16
NU3	HSM_MODE2	MODCLKPOL2	MMODE2			PMODE2	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
NU2	HSM_MODE1	MODCLKPOL1	MMODE1			PMODE1	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	
7	6	5	4	3	2	1	0
NU1	HSM_MODE0	MODCLKPOL0	MMODE0			PMODE0	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

Table 23-37. SPIPMCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
30	HSM_MODE3	R/W	0h	High Speed Modulo Mode control bit for Data Format 3. Controls whether the PMODE3 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE3 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE3 bits are non-zero
29	MODCLKPOL3	R/W	0h	Modulo mode SPICLK Polarity for Data Format 3 Determines the Polarity of the SPICLK in Modulo mode only. If MODULO MODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected
28-26	MMODE3	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 3.000 = Normal single dataline mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
25-24	PMODE3	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 3. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”) 10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)
23	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
22	HSM_MODE2	R/W	0h	High Speed Modulo Mode control bit for Data Format 2. Controls whether the PMODE2 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE2 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE2 bits are non-zero

Table 23-37. SPIPMCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	MODCLKPOL2	R/W	0h	Modulo mode SPICLK Polarity for Data Format 2. Determines the Polarity of the SPICLK in Modulo mode only. If MMODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected
20-18	MMODE2	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 2. 000 = 1-data line Mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
17-16	PMODE2	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 2. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”)10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)
15	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
14	HSM_MODE1	R/W	0h	High Speed Modulo Mode control bit for Data Format 1. Controls whether the PMODE1 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE1 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE1 bits are non-zero.
13	MODCLKPOL1	R/W	0h	Modulo mode SPICLK Polarity for Data Format 1. Determines the Polarity of the SPICLK in Modulo mode only. If MMODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected.
12-10	MMODE1	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 1. 000 = 1-data line Mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
9-8	PMODE1	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 1. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”) 10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)
7	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
6	HSM_MODE0	R/W	0h	High Speed Modulo Mode control bit for Data Format 0. Controls whether the PMODE0 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE0 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE0 bits are non-zero.
5	MODCLKPOL0	R/W	0h	Modulo mode SPICLK Polarity for Data Format 0. Determines the Polarity of the SPICLK in Modulo mode only. If MMODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected.

Table 23-37. SPIPMCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-2	MMODE0	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 0. 000 = 1-data line Mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
1-0	PMODE0	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 0. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”) 10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)

23.3.1.27 MIBSPIE Register (Offset = 70h) [reset = 500h]

MIBSPIE is shown in Figure 23-124 and described in Table 23-106.

Return to Summary Table.

MibSPI Enable Register

Figure 23-58. MIBSPIE Register

31	30	29	28	27	26	25	24
NU3							
R-0h							
23	22	21	20	19	18	17	16
NU3							RXRAMACCESS
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU2				EXTENDED_BUF_ENA			
R-0h				R/W-5h			
7	6	5	4	3	2	1	0
NU1							MSPIENA
R-0h							R/W-0h

Table 23-38. MIBSPIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
16	RXRAMACCESS	R/W	0h	Receive RAM Access control Bit. During normal operating mode of MibSPI, the Receive Data/Status portion of Multibuffer RAM is read-only. To enable testing of Data Integrity checks of Receive RAM, a special read/write access control is provided through this bit. 0 = The RX portion of Multibuffer RAM is not writable by the CPU. That is, portion of Multibuffer RAM, addressed by an offset of 0x200-0x3FF is write protected. 1 = The whole of Multibuffer RAM is fully accessible for read/write by the CPU.
15-12	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect
11-8	EXTENDED_BUF_ENA	R/W	5h	Enables the support for 256 buffers. By default MibSPI supports up to 128 buffers for both TX and RX. It is also possible to extend the support to 256 buffers as a parameterized implementation. This field can be used to enable/disable the support for Extended Buffers. This Enable field is implemented only if “EXTENDED_BUF” parameter is set to ‘1’. If the parameter is set to ‘0’, this field is read-only and reads the disable value. Write (Privilege mode only) 1010 - Enable the Extended Buffer mode - up to 256 buffers can be used 0101 - Disable the Extended Buffer mode - MibSPI supports only 128 buffers All other values - writes are ignored and the values are not updated into this field. The state of the feature remains unchanged. Read (both privilege and user modes) 1010 - Extended Buffer mode is enabled - up to 256 buffers can be used 0101 - Extended Buffer mode is disabled - MibSPI supports only 128 buffers
7-1	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.

Table 23-38. MIBSPIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	MSPIENA	R/W	0h	Multibuffer mode Enable. After power-up or reset MSPIENA remains cleared, which means that the MibSPI runs in compatibility mode by default. If Multibuffer mode is desired, this register should be configured first after configuring the SPIGCR0 register. Unless MSPIENA is set to '1', the Multibuffer mode registers are not writable. Refer to Section 3.3 for the grouping of registers into Compatibility mode and Multibuffer mode. 1 =The MibSPI is configured to run in MibSPI mode (Multibuffer mode). In this mode the additional features are available. 0 =The MibSPI runs in compatibility mode, i.e. in this mode the MibSPI is fully code compliant to the standard TMS470 Platform SPI. No Multibuffer feature is supported

23.3.1.28 TGITENST Register (Offset = 74h) [reset = 0h]

TGITENST is shown in Figure 23-125 and described in Table 23-107.

Return to Summary Table.

MibSPI Transfer Group Interrupt Enable Set Register

Figure 23-59. TGITENST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETINTENRDY																SETINTENSUS															
R/W-0h																R/W-0h															

Table 23-39. TGITENST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SETINTENRDY	R/W	0h	Transfer group interrupt set (enable) when transfer finished. Write: 1 = Enables the “The Transfer group x completed “ interrupt Interrupt gets generated when Transfer Group x gets completed. 0 = Has no effect. Read: 1 = “The Transfer group x completed “ interrupt is enabled Interrupt gets generated when Transfer Group x gets completed. 0 = “The Transfer group x completed” interrupt is disabled Interrupt does not get generated when Transfer Group x gets completed
15-0	SETINTENSUS	R/W	0h	Transfer group interrupt set (enable) when transfer suspended Write: 1 = Enables the “The Transfer group x suspended “ interrupt Interrupt gets generated when Transfer Group x gets suspended. 0 = Has no effect. Read: 1 = “The Transfer group x suspended “ interrupt is enabled Interrupt gets generated when Transfer Group x gets suspended. 0 = “The Transfer group x suspended” interrupt is disabled Interrupt does not get generated when Transfer Group x gets suspended

23.3.1.29 TGITENCR Register (Offset = 78h) [reset = 0h]

TGITENCR is shown in Figure 23-126 and described in Table 23-108.

Return to Summary Table.

MibSPI Transfer Group Interrupt Enable Clear Register

Figure 23-60. TGITENCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRINTENRDY																CLRINTENSUS															
R/W-0h																R/W-0h															

Table 23-40. TGITENCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CLRINTENRDY	R/W	0h	Transfer group interrupt clear (disable) when transfer finished. Write: 1 = Disables the “The Transfer group x completed “ interrupt Interrupt does not get generated when Transfer Group x gets completed. 0 = Has no effect. Read: 1 = “The Transfer group x completed “ interrupt is enabled Interrupt gets generated when Transfer Group x gets completed. 0 = “The Transfer group x completed” interrupt is disabled Interrupt does not get generated when Transfer Group x gets completed
15-0	CLRINTENSUS	R/W	0h	Transfer group interrupt clear (disable) when transfer suspended Write: 1 = Disables the “The Transfer group x suspended “ interrupt Interrupt does not get generated when Transfer Group x gets suspended. 0 = Has no effect. Read: 1 = “The Transfer group x suspended “ interrupt is enabled Interrupt gets generated when Transfer Group x gets suspended. 0 = “The Transfer group x suspended” interrupt is disabled Interrupt does not get generated when Transfer Group x gets suspended

23.3.1.30 TGITLVST Register (Offset = 7Ch) [reset = 0h]

TGITLVST is shown in Figure 23-127 and described in Table 23-109.

Return to Summary Table.

MibSPI Transfer Group Interrupt Level Set Register

Figure 23-61. TGITLVST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETINTLVLRDY																SETINTLVLSUS															
R/W-0h																R/W-0h															

Table 23-41. TGITLVST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SETINTLVLRDY	R/W	0h	Transfer group completed" Interrupt Level set register Write: 1 = Sets the "The Transfer group x completed " interrupt to line INT1 0 = Has no effect. Read: 1 = "The Transfer group x completed " interrupt is set to line INT1 0 = "The Transfer group x completed " interrupt is set to line INT0
15-0	SETINTLVLSUS	R/W	0h	Transfer group suspended" interrupt Level set rigester Write: 1 = Sets the "The Transfer group x suspended " interrupt to line INT1 0 = Has no effect. Read: 1 = "The Transfer group x suspended " interrupt is set to line INT1 0 = "The Transfer group x suspended " interrupt is set to line INT0

23.3.1.31 TGITLVCR Register (Offset = 80h) [reset = 0h]

TGITLVCR is shown in Figure 23-128 and described in Table 23-110.

Return to Summary Table.

MibSPI Transfer Group Interrupt Level Clear Register

Figure 23-62. TGITLVCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRINTLVLRDY																CLRINTLVLSUS															
R/W-0h																R/W-0h															

Table 23-42. TGITLVCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CLRINTLVLRDY	R/W	0h	Transfer group completed" Interrupt Level clear register Write: 1 = Sets the "The Transfer group x completed " interrupt to line INT0 0 = Has no effect. Read: 1 = "The Transfer group x completed " interrupt is set to line INT1 0 = "The Transfer group x completed " interrupt is set to line INT0
15-0	CLRINTLVLSUS	R/W	0h	Transfer group suspended" interrupt Level clear register Write: 1 = Sets the "The Transfer group x suspended " interrupt to line INT0 0 = Has no effect. Read: 1 = "The Transfer group x suspended " interrupt is set to line INT1 0 = "The Transfer group x suspended " interrupt is set to line INT0

23.3.1.32 TGINTFLAG Register (Offset = 84h) [reset = 0h]

TGINTFLAG is shown in Figure 23-129 and described in Table 23-111.

Return to Summary Table.

Transfer Group Interrupt Flag Register

Figure 23-63. TGINTFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTFLGRDY																INTFLGSUS															
R-0h																R-0h															

Table 23-43. TGINTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	INTFLGRDY	R	0h	Transfer group interrupt flag for “transfer finished” interrupt. Read: 1 =A “transfer finished” interrupt from transfer group x occurred. No matter whether the interrupt is enabled or disabled (INTENRDYx = don’t care) or whether the interrupt is mapped to line INT0 or INT1, INTFLGRDYx is set right after the transfer from transfer group x is finished. 0 =No “transfer finished” interrupt occurred since last clearing of the flag INTFLGRDYx. Write: 1 = Clears the corresponding bit flag. 0 = Has no effect.
15-0	INTFLGSUS	R	0h	Transfer group interrupt flag for “transfer suspend” interrupt. Read: 1 =A “transfer suspended” interrupt from transfer group x occurred. No matter whether the interrupt is enabled or disabled (INTENSUSx = don’t care) or whether the interrupt is mapped to line INT0 or INT1, INTFLGSUSx is set right after the transfer from transfer group x is suspended. 0 =No ”transfer suspended” interrupt occurred since last clearing of the flag INTFLGSUSx. Note: Read Clear Behavior Reading the interrupt vector registers TGINTVECT0 or TGINTVECT1 automatically clears the interrupt flag bit INTFLGRDYx referenced by the vector number given by INTVECT0/INTVECT1 bits, if SUSPEND0/SUPEND1 bit in the Vector registers is ‘0’.

23.3.1.33 TICKCNT Register (Offset = 90h) [reset = 0h]

TICKCNT is shown in Figure 23-130 and described in Table 23-112.

Return to Summary Table.

Tick Count Register

Figure 23-64. TICKCNT Register

31	30	29	28	27	26	25	24
TICKENA	RELOAD	CLKCTRL		NU			
R/W-0h	R-0/W-0h	R/W-0h		R-0h			
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
TICKVALUE							
R/W-0h							
7	6	5	4	3	2	1	0
TICKVALUE							
R/W-0h							

Table 23-44. TICKCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TICKENA	R/W	0h	Tick counter enable. 1 =The MibSPI internal tick counter is enabled and is clocked by the clock source selected by CLKCTRL[1:0]. When the tick counter is enabled it starts down-counting from its current value. When TICKENA goes from “0” to “1” the tick counter is automatically loaded with the TICKVALUE. 0 =The MibSPI internal tick counter is disabled. The counter value remains unchanged. Note: When the tick counter is disabled the trigger signal is forced low.
30	RELOAD	R-0/W	0h	Re-load tick counter. RELOAD is a set-only bit, i.e. writing a “1” to it automatically reloads the Tick Counter with the value stored in TICKVALUE. Reading RELOAD always returns a “0”. Note: When the tick counter is reloaded by the RELOAD bit, the trigger signal is not toggled.
29-28	CLKCTRL	R/W	0h	Tick counter clock source control. CLKCTRL[1:0] defines the clock source that is used to clock the MibSPI internal tick counter.CLKCTRL[1:0] Description 00b SPICLK of Data word format 0 is selected as clock source of tick counter 01b SPICLK of Data word format 1 is selected as clock source of tick counter 10b SPICLK of Data word format 2 is selected as clock source of tick counter 11b SPICLK of Data word format 3 is selected as clock source of tick counter
27-16	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
15-0	TICKVALUE	R/W	0h	Initial value for tick counter. TICKVALUE stores the initial value for the tick counter. The tick counter is loaded with TICKVALUE every time an under-flow condition occurs and every time the RELOAD flag is set by the host

23.3.1.34 LTGPEND Register (Offset = 94h) [reset = 0h]

LTGPEND is shown in Figure 23-131 and described in Table 23-113.

Return to Summary Table.

Last Transfer Group End Pointer

Figure 23-65. LTGPEND Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3			TGINSERVICE					NU2							
R-0h			R-0h					R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPEND								NU1							
R/W-0h								R-0h							

Table 23-45. LTGPEND Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU3	R	0h	Reserved.Reads return ‘0’ and writes have no effect
28-24	TGINSERVICE	R	0h	Transfer Group currently being serviced by the Sequencer. Read-Only field indicating the current Transfer Group that is being serviced. This field can generally be used for code debug purpose. Read Value:TG IN SERVICE[4:0] Description 00000b No Transfer Group is being serviced by the Sequencer 00001b Transfer Group0 is being serviced by the Sequencer ... ... 10000b Transfer Group15 is being serviced by the Sequencer 10001b - 11111b Invalid values
23-16	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect
15-8	LPEND	R/W	0h	Last Transfer Group End Pointer Usually the transfer group end address (PEND) is inherently defined by the start value of the starting pointer of the subsequent transfer group (PSTART). The transfer group ends at the buffer one before the next transfer group starts (PEND[x]=PSTART[x+1] - 1). For a full configuration of MibSPI, the 15th transfer group has no subsequent transfer group, i.e. no end address is inherently defined. Therefore LPEND has to be programmed to specify explicitly the end address of the 15th transfer group. Number of Transfer Groups implemented in a MibSPI can vary from one MibSPI to another since it is a generic parameter based implementation. If in a MibSPI, only 4 Transfer Groups are implemented, then the PEND of the 4th TG is defined by LPEND values defined in this LTGPEND register.
7-0	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect

23.3.1.35 TG0CTRL Register (Offset = 98h) [reset = 0h]

TG0CTRL is shown in Figure 23-132 and described in Table 23-114.

Return to Summary Table.

MibSPI Transfer Group Control Register The number of transfer groups is scalable by design up to a maximum of 16. Depending on the implementation the number of transfer groups and hence the number of transfer group control register may vary. Each transfer group can be configured via one dedicated control register. The register description below shows one exemplary control register(x) which is identical for all transfer groups. E.g. the control register for transfer group 2 is named “TG2CTRL” and is located at address base0+98h+4*2.

Figure 23-66. TG0CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-46. TG0CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.

Table 23-46. TG0CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved

Table 23-46. TG0CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.36 TG1CTRL Register (Offset = 9Ch) [reset = 0h]

TG1CTRL is shown in Figure 23-133 and described in Table 23-115.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-67. TG1CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-47. TG1CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-47. TG1CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-47. TG1CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.37 TG2CTRL Register (Offset = A0h) [reset = 0h]

TG2CTRL is shown in Figure 23-134 and described in Table 23-116.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-68. TG2CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-48. TG2CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-48. TG2CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-48. TG2CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.38 TG3CTRL Register (Offset = A4h) [reset = 0h]

TG3CTRL is shown in Figure 23-135 and described in Table 23-117.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-69. TG3CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-49. TG3CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-49. TG3CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-49. TG3CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.39 TG4CTRL Register (Offset = A8h) [reset = 0h]

TG4CTRL is shown in Figure 23-136 and described in Table 23-118.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-70. TG4CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-50. TG4CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-50. TG4CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-50. TG4CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.40 TG5CTRL Register (Offset = ACh) [reset = 0h]

TG5CTRL is shown in Figure 23-137 and described in Table 23-119.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-71. TG5CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-51. TG5CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-51. TG5CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-51. TG5CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.41 TG6CTRL Register (Offset = B0h) [reset = 0h]

TG6CTRL is shown in Figure 23-138 and described in Table 23-120.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-72. TG6CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-52. TG6CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-52. TG6CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-52. TG6CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.42 TG7CTRL Register (Offset = B4h) [reset = 0h]

TG7CTRL is shown in Figure 23-139 and described in Table 23-121.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-73. TG7CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-53. TG7CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-53. TG7CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-53. TG7CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.1.43 DMA0CTRL Register (Offset = D8h) [reset = 0h]

DMA0CTRL is shown in Figure 23-140 and described in Table 23-122.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-74. DMA0CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-54. DMA0CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-54. DMA0CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.1.44 DMA1CTRL Register (Offset = DCh) [reset = 0h]

DMA1CTRL is shown in Figure 23-141 and described in Table 23-123.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-75. DMA1CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-55. DMA1CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-55. DMA1CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.1.45 DMA2CTRL Register (Offset = E0h) [reset = 0h]

DMA2CTRL is shown in Figure 23-142 and described in Table 23-124.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-76. DMA2CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-56. DMA2CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-56. DMA2CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.1.46 DMA3CTRL Register (Offset = E4h) [reset = 0h]

DMA3CTRL is shown in Figure 23-143 and described in Table 23-125.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-77. DMA3CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-57. DMA3CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-57. DMA3CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.1.47 DMA4CTRL Register (Offset = E8h) [reset = 0h]

DMA4CTRL is shown in Figure 23-144 and described in Table 23-126.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-78. DMA4CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-58. DMA4CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-58. DMA4CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.1.48 ICOUNT0 Register (Offset = F8h) [reset = 0h]

ICOUNT0 is shown in Figure 23-145 and described in Table 23-127.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-79. ICOUNT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-59. ICOUNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.1.49 ICOUNT1 Register (Offset = FCh) [reset = 0h]

ICOUNT1 is shown in Figure 23-146 and described in Table 23-128.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-80. ICOUNT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-60. ICOUNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.1.50 ICOUNT2 Register (Offset = 100h) [reset = 0h]

ICOUNT2 is shown in Figure 23-147 and described in Table 23-129.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-81. ICOUNT2 Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT															
R-0h															

Table 23-61. ICOUNT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.1.51 ICOUNT3 Register (Offset = 104h) [reset = 0h]

ICOUNT3 is shown in Figure 23-148 and described in Table 23-130.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-82. ICOUNT3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-62. ICOUNT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.1.52 ICOUNT4 Register (Offset = 108h) [reset = 0h]

ICOUNT4 is shown in Figure 23-149 and described in Table 23-131.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-83. ICOUNT4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-63. ICOUNT4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.1.53 DMACNTLEN Register (Offset = 118h) [reset = 0h]

DMACNTLEN is shown in Figure 23-150 and described in Table 23-132.

Return to Summary Table.

DMA LARGE COUNT register

Figure 23-84. DMACNTLEN Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							LARGE_COUNT
R-0h							R/W-0h

Table 23-64. DMACNTLEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
0	LARGE_COUNT	R/W	0h	0: Writes to the DMAxCTRL register will modify the ICOUNT value. Reading ICOUNT and COUNT can be done from the DMAxCTRL register. The DMAxCOUNT register should not be used since any write to this register will be overwritten by a subsequent write to DMAxCTRL register to set the TXDMAENA or RXDMAENA bits. 1: Writes to the DMAxCTRL register will not modify the ICOUNT value. The ICOUNT value must be written to in the DMAxCOUNT register before the RXDMAENA or TXDMAENA bits are set in the DMAxCTRL register. The DMAxCOUNT register should be used for reading COUNT or ICOUNT.

23.3.1.54 PAR_ECC_CTRL Register (Offset = 120h) [reset = 050A0005h]

PAR_ECC_CTRL is shown in Figure 23-151 and described in Table 23-133.

Return to Summary Table.

Parity/ECC Control Register

Figure 23-85. PAR_ECC_CTRL Register

31	30	29	28	27	26	25	24
NU4				SBE_EVT_EN			
R-0h				R/W-5h			
23	22	21	20	19	18	17	16
NU3				EDAC_MODE			
R-0h				R/W-Ah			
15	14	13	12	11	10	9	8
NU2							PTESTEN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NU1				EDEN			
R-0h				R/W-5h			

Table 23-65. PAR_ECC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU4	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
27-24	SBE_EVT_EN	R/W	5h	Single Bit Error Event Enable This bit controls the generation of Error signaling (on MIBSPI_SBERR port) whenever a Single Bit Errors (SBE) is detected on TXRAM/RXRAM. This signal can be used to generate interrupt if required. Write: 0101 - Disable Error Event indication upon detection of SBE on TXRAM/RXRAM 1010 - Enable Error Event upon detection of SBE on TXRAM/RXRAM All other values - writes are ignored and the values are not updated into this field. The state of the feature remains unchanged. Read: Returns the current value of the field
23-20	NU3	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
19-16	EDAC_MODE	R/W	Ah	Error Detection And Correction Mode These bits determine whether Single Bit Errors (SBE) detected by the SECDED block will be corrected or not. Write: 0101 - Disable correction of SBE detected by the SECDED block 1010 - Enable correction of SBE detected by the SECDED block All other values - writes are ignored and the values are not updated into this field. The state of the feature remains unchanged. Read: Returns the current value of the field
15-9	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
8	PTESTEN	R/W	0h	Parity/ECC memory Test Enable. This bit, maps the parity/ecc bits corresponding to Multibuffer RAM locations into the peripheral RAM frame to make them accessible by the CPU. User and privilege mode (read): 0 = parity/ecc bits are not memory mapped 1 = parity/ecc bits are memory mapped Privilege mode (write): 0 = disable memory mapping of Parity/ECC locations 1 = enable memory mapping of Parity/ECC locations
7-4	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
3-0	EDEN	R/W	5h	Error Detection Enable These bits enable Parity/ECC Error Detection. Write: 0101: Disables Parity/ECC Error Detection Logic(default) Others : Enables Parity/ECC Error Detection Logic. Read: Returns the current value of this field

23.3.1.55 PAR_ECC_STAT Register (Offset = 124h) [reset = 0h]

PAR_ECC_STAT is shown in Figure 23-152 and described in Table 23-134.

Return to Summary Table.

Parity/ECC Status Register

Figure 23-86. PAR_ECC_STAT Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2							
R-0h							
15	14	13	12	11	10	9	8
NU2						SBE_FLG1	SBE_FLG0
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
NU1						UERR_FLG1	UERR_FLG0
R-0h						R-0h	R-0h

Table 23-66. PAR_ECC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
9	SBE_FLG1	R	0h	Single Bit Error in RXRAM. This flag indicates if a single bit ECC Error ocurred on reading RXRAM Read: 0 = No error occured. 1 = Single bit error is detected in RXRAM and the address is captured in SBERRADDR1 register. Write: 0 = No effect. 1 = Clears the bit.
8	SBE_FLG0	R	0h	Single Bit Error in TXRAM. This flag indicates if a single bit ECC Error ocurred on reading TXRAM Read: 0 = No error occured. 1 = Single bit error is detected in TXRAM and the address is captured in SBERRADDR0 register. Write: 0 = No effect. 1 = Clears the bit.
7-2	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
1	UERR_FLG1	R	0h	Uncorrectable Parity or double bit ECC error detection flag This flag indicates if a Parity or double bit ECC error ocurred on reading RXRAM When this bit is read: 0 = No error occured. 1 = Error detected and the address is captured in UERRADDR1 register. When write to this bit with: 0 = No effect. 1 = Clears the bit
0	UERR_FLG0	R	0h	Uncorrectable Parity or double bit ECC error detection flag This flag indicates if a Parity or ECC error ocurred on reading TXRAM When this bit is read: 0 = No error occured. 1 = Error detected and the address is captured in UERRADDR0 register. When write to this bit with: 0 = No effect. 1 = Clears the bit.

23.3.1.56 UERRADDR1 Register (Offset = 128h) [reset = 0h]

UERRADDR1 is shown in Figure 23-153 and described in Table 23-135.

Return to Summary Table.

Uncorrectable Parity or double bit ECC error Address Register - RXRAM

Figure 23-87. UERRADDR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																					UERRADDR1										
R-0h																					R-0h										

Table 23-67. UERRADDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	UERRADDR1	R	0h	Uncorrectable Parity or double bit ECC error address This register holds the address of the RAM location if a parity or double bit ECC error is detected when reading the MibSPI (Receive) RXRAM. The address captured is byte alligned when RAM Parity Check is supported. This error address is frozen from being updated until it is read by the VBUS host. Reading this register clears its contents to the default value The default value is 0x400 if Extended Buffer feature is enabled, else it is 0x200 Writes to this register are ignored

23.3.1.57 UERRADDR0 Register (Offset = 12Ch) [reset = 0h]

UERRADDR0 is shown in Figure 23-154 and described in Table 23-136.

Return to Summary Table.

Uncorrectable Parity or double bit ECC error address register - TXRAM

Figure 23-88. UERRADDR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																					UERRADDR0										
R-0h																					R-0h										

Table 23-68. UERRADDR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	UERRADDR0	R	0h	Uncorrectable Parity or double bit ECC error address This register holds the address when a parity error is generated while reading the MibSPI (Transmit) TXRAM. The TXRAM can be read either by CPU or by the MibSPI Sequencer FSM logic for transmission. The address captured is byte alligned. This error address is frozen from being updated until it is read by the VBUSP host. Reading this register clears its contents to the default value of 0x000. Writes to this register are ignored.

23.3.1.58 RXOVRN_BUF_ADDR Register (Offset = 130h) [reset = 200h]

RXOVRN_BUF_ADDR is shown in Figure 23-155 and described in Table 23-137.

Return to Summary Table.

Receive RAM Overrun Buffer Address Register

Figure 23-89. RXOVRN_BUF_ADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU					RXOVRN_BUF_ADDR										
R-0h					R-200h										

Table 23-69. RXOVRN_BUF_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	RXOVRN_BUF_ADDR	R	200h	Address of the RAM location of RXRAM for which an Overwrite occured. This address value will show only the offset address of the RAM location in the Multibuffer RAM address space. Refer to the device Spec for the actual absolute address of RXRAM. Content of this register are valid only when any of the TGINTVECT0 or TGINTVECT1 and SPIFLG registers show an RXOVRN error vector while in Multibuffer mode. If there are multiple Overrun errors, then this register holds address of first overrun address until it is read.

23.3.1.59 IOLPBKTSTCR Register (Offset = 134h) [reset = 0h]

IOLPBKTSTCR is shown in Figure 23-156 and described in Table 23-138.

Return to Summary Table.

SPI/MibSPI IO Loopback Test Control Register This register controls test mode for I/O pins. It also controls whether loop-back should be digital or analog ones in this test mode. In addition it contains control bits to induce some of the error condition into the module. These are to be used for test purpose only. All the control/status bits in this register are valid only when IO LPBK TST ENA field is set to “1010”.

Figure 23-90. IOLPBKTSTCR Register

31	30	29	28	27	26	25	24
NU4							SCSFAILFLG
R-0h							0h
23	22	21	20	19	18	17	16
NU3			CTRLBITERR	CTRLDESYNC	CTRLPARERR	CTRLTIMEOUT	CTRLDLENERR
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
NU2				IOLPBKTSTENA			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
NU1			ERRSCSPIN		CTRLSCSPINERR	LPBKTYPE	RXPENA
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 23-70. IOLPBKTSTCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	SCSFAILFLG		0h	Bit indicating a failure on SPISCS pin compare during analog loopback during IO Loopback Test mode. Read 1 = A comparison between the internal CSNR field and the analog looped back value of SPISCS[7:0] pins failed. A stuck-at fault is detected on one of the SPISCS[7:0]. Comparison is done only on the pins which are configured as functional and during transfer operation. 0 = No miscompares on any of the 8 chipselect pin value comparison with the internal Chipselect number CSNR during transfers. Write 1 = Clear this Flag bit. 0 = No effect.
23-21	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
20	CTRLBITERR	R/W	0h	Controls inducing of BITERR during IO Loopback Test mode. 1 = The value of incoming data from the loopback Transmit pin is flipped. 0 = No affect on BIT ERROR.
19	CTRLDESYNC	R/W	0h	Controls inducing of DESYNC Error during IO Loopback Test mode. 1 = Forces the incoming SPIENA pin (if functional) to remain ‘0’ even after the transfer complete. This forcing will be retained until the Kernel reaches IDLE state. 0 = No affect on DESYNC Error.
18	CTRLPARERR	R/W	0h	Controls inducing of Parity Error during IO Loopback Test mode. 1 = Flips the Parity Polarity signal being used for transmit parity generation logic 0 = No affect on Parity Error
17	CTRLTIMEOUT	R/W	0h	Controls inducing of TIMEOUT Error during IO LoopbacK Test mode. 1 = Forces the incoming SPIENA pin (if functional) to remain ‘1’ when transmission is initiated. The forcing will be retained until the Kernel reaches IDLE state. 0 = No affect on TIMEOUT Error.

Table 23-70. IOLPBKTSTCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CTRLDLENERR	R/W	0h	Controls inducing of Data Length Error during IO Loopback Test mode. 1 = When in Master mode, forces the SPIENA pin(if functional) to '1' when the module starts Shifting the data. When in Slave mode, forces the incoming SPISCS pin(if functional) to '1' when the module starts shifting the data.. 0 = No affect on Data Length Error.
15-12	NU2	R	0h	Reserved. Reads return '0' and writes have no effect.
11-8	IOLPBKTSTENA	R/W	0h	Module I/O Loopback Test Enable Key User and Privileged mode reads. Write access only in Privileged mode. Write: 1010 = I/O DFT is enabled All other values = I/O DFT is disabled Read: 1010 = I/O DFT is enabled All other values = I/O DFT is disabled
7-6	NU1	R	0h	Reserved. Reads return '0' and writes have no effect.
5-3	ERRSCSPIN	R/W	0h	Inject Error on ChipSelect Pin. The value in this field is decoded to find out the ChipSelect pin on which to inject an error. During the analog loopback of IO Loopback Test mode if CTRL SCS PIN ERR bit is set to '1', then the chipselect pin selected by this field is forced to the opposite of its original CSNR bit. 000 - Select SPISCS[0] for injecting error 001 - Select SPISCS[1] for injecting error . 111 - Select SPISCS[7] for injecting error
2	CTRLSCSPINERR	R/W	0h	Control bit to enable the injection of an error on SPISCS[7:0] pins. Individual pins of SPISCS[7:0] can be choosen using ERR SCS PIN. 1 = Enable the error inducing logic to the SPISCS pins. 0 = Disable the error inducing logic.
1	LPBKTYPE	R/W	0h	Module IO Loopback Type (Analog/Digital). User and Privileged mode reads. Write access only in Privileged mode. Write/Read : 1 = Analog loopback is enabled in module I/O DFT mode when IOLPBKTSTENA = 1010) 0 = Digital loopback is enabled in module I/O DFT mode when IOLPBKTSTENA = 1010
0	RXPENA	R/W	0h	Module Analog loopback through Receive Pin Enable. User and Privileged mode reads. Write only in privileged mode: Write/Read : 1 = Analog loopback through receive pin 0 = Analog loopback through transmit pin. This bit is valid only when LPBK TYPE = '1' which chooses Analog loopback mode.

23.3.1.60 EXTENDED_PRESCALE1 Register (Offset = 138h) [reset = 0h]

EXTENDED_PRESCALE1 is shown in Figure 23-157 and described in Table 23-139.

Return to Summary Table.

SPI/MibSPI Extended Prescale Register 1 (EXTENDED_PRESCALE1 for SPIFMT0 and SPIFMT1) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This is an extension of SPIFMT0 and SPIFMT1 registers. For example, EPRESCALE_FMT1(7:0) of EXTENDED_PRESCALE1 and PRESCALE1(7:0) of SPIFMT1 register will always reflect the same contents. Similarly EPRESCALE_FMT0(7:0) and PRESCALE0(7:0) of SPIFMT0 reflect the same contents.

Figure 23-91. EXTENDED_PRESCALE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					EPRESCLAE_FMT1										
R-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					EPRESCLAE_FMT0										
R-0h					R/W-0h										

Table 23-71. EXTENDED_PRESCALE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
26-16	EPRESCLAE_FMT1	R/W	0h	Extended Prescale value for SPIFMT1. EPRESCALE_FMT1 can be modified in privilege mode only. EPRESCALE_FMT1 determines the bit transfer rate of Data Format 1 if the SPI/MibSPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT1(7:0) bits and PRESCALE1(7:0) bits of SPIFMT1 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT1(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EXTENDED_PRESCALE1(26:16) or SPIFMT1(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE1 register is programmed after SPIFMT1 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE1 register when a prescale of more 256 is intended on SPICLK. BRFormatx = VBUSPCLK/(EXTENDEDPRESCALEy+1) When EPRESCALE_FMTy (y=1,2) is set to zero(0), the SPI clock rate defaults to VBUSPCLK/2.
15-11	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.

Table 23-71. EXTENDED_PRESCALE1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-0	EPRESCLAE_FMT0	R/W	0h	EPRESCALE_FMT0 can be modified in privilege mode only. EPRESCALE_FMT0 determines the bit transfer rate of Data Format 0 if the SPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT0(7:0) bits and PRESCALE0(7:0) bits of SPIFMT0 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT0(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EPRESCALE0(26:16) or SPIFMT0(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE1 register is programmed after SPIFMT0 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE1 register when a prescale of more 256 is intended on SPICLK.

23.3.1.61 EXTENDED_PRESCALE2 Register (Offset = 13Ch) [reset = 0h]

EXTENDED_PRESCALE2 is shown in Figure 23-158 and described in Table 23-140.

Return to Summary Table.

SPI/MibSPI Extended Prescale Register 2 (EXTENDED_PRESCALE2 for SPIFMT2 and SPIFMT3) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This register is an extension of SPIFMT2 and SPIFMT3 registers. For example, EPRESCALE_FMT2(7:0) of EXTENDED_PRESCALE2 and PRESCALE2(7:0) of SPIFMT2 register will always reflect the same contents. Similarly EPRESCALE_FMT3(7:0) and PRESCALE3(7:0) of SPIFMT3 reflect the same contents.

Figure 23-92. EXTENDED_PRESCALE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU4					EPRESCLAE_FMT3										
R-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3					EPRESCLAE_FMT2										
R-0h					R/W-0h										

Table 23-72. EXTENDED_PRESCALE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
26-16	EPRESCLAE_FMT3	R/W	0h	EPRESCALE_FMT3 can be modified in privilege mode only. EPRESCALE_FMT3 determines the bit transfer rate of Data Format 3 if the SPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT3(7:0) bits and PRESCALE3(7:0) bits of SPIFMT3 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT3(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EPRESCALE3(26:16) or SPIFMT3(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE2 register is programmed after SPIFMT3 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE2 register when a prescale of more 256 is intended on SPICLK.
15-11	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
10-0	EPRESCLAE_FMT2	R/W	0h	EPRESCALE_FMT2 can be modified in privilege mode only. EPRESCALE_FMT2 determines the bit transfer rate of Data Format 2 if the SPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT2(7:0) bits and PRESCALE2(7:0) bits of SPIFMT2 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT3(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EXTENDED_PRESCALE2(26:16) or SPIFMT2(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE2 register is programmed after SPIFMT2 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE2 register when a prescale of more 256 is intended on SPICLK.

23.3.1.62 ECCDIAG_CTRL Register (Offset = 140h) [reset = Ah]

ECCDIAG_CTRL is shown in Figure 23-159 and described in Table 23-141.

Return to Summary Table.

ECC Diagnostic Control register

Figure 23-93. ECCDIAG_CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU												ECCDIAG_EN			
R-0h												R/W-Ah			

Table 23-73. ECCDIAG_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
3-0	ECCDIAG_EN	R/W	Ah	ECC Diagnostic mode Enable Key bits. 0101 : Diagnostic mode is enabled. Writes and reads from ECC bits allowed from the ECC address space. Refer to Section 9 for details on ECC/Parity address space. Others : Diagnostic mode is disabled. No writes to ECC bits are ignored, reads return ‘0’.

23.3.1.63 ECCDIAG_STAT Register (Offset = 144h) [reset = 0h]

ECCDIAG_STAT is shown in Figure 23-160 and described in Table 23-142.

Return to Summary Table.

ECC Diagnostic Status register

Figure 23-94. ECCDIAG_STAT Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2						DEFLG1	DEFLG0
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
NU1							
R-0h							
7	6	5	4	3	2	1	0
NU1						SEFLG1	SEFLG0
R-0h						R-0h	R-0h

Table 23-74. ECCDIAG_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
17	DEFLG1	R	0h	Double bit error flag for RXRAM 1 - A double bit Error is detected for RXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.
16	DEFLG0	R	0h	Double bit error flag for TXRAM 1 - A double bit Error is detected for TXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.
15-2	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
1	SEFLG1	R	0h	Single bit error flag for RXRAM 1 - A Single bit Error is detected for RXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.
0	SEFLG0	R	0h	Single bit error flag for TXRAM 1 - A Single bit Error is detected for TXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.

23.3.1.64 SBERRADDR1 Register (Offset = 148h) [reset = 0h]

SBERRADDR1 is shown in Figure 23-161 and described in Table 23-143.

Return to Summary Table.

Single Bit Error Address Register - RXRAM

Figure 23-95. SBERRADDR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																					SBERRADDR1										
R-0h																					R-0h										

Table 23-75. SBERRADDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	SBERRADDR1	R	0h	Single Bit ECC Error Address This register holds the address of the RAM location when a single bit error is generated by SECDED block while reading the MibSPI (Receive) RXRAM. This error address is frozen from being updated until it is read by the VBUS host. Reading this register clears its contents to the default value The default value is 0x400 if Extended Buffer feature is enabled, else it is 0x200 Writes to this register are ignored.

23.3.1.65 SBERRADDR0 Register (Offset = 14Ch) [reset = 0h]

SBERRADDR0 is shown in Figure 23-162 and described in Table 23-144.

Return to Summary Table.

Single Bit ECC Error Address Register - TXRAM

Figure 23-96. SBERRADDR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2																					SBERRADDR0										
R-0h																					R-0h										

Table 23-76. SBERRADDR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	SBERRADDR0	R	0h	Single Bit ECC Error Address This register holds the address when a single bit error is generated from SECDED block while reading the MibSPI (Transmit) TXRAM. The TXRAM can be read either by CPU or by the MibSPI Sequencer logic for transmission. This error address is frozen from being updated until it is read by the VBUSP host. Reading this register clears its contents to the default value of 0x000. Writes to this register are ignored.

23.3.1.66 SPIREV Register (Offset = 1FCh) [reset = 4A050308h]

SPIREV is shown in Figure 23-163 and described in Table 23-145.

Return to Summary Table.

SPI / MibSPI Revision ID Register

Figure 23-97. SPIREV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		NU		FUNC											
R-1h		R-0h		R-A05h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-0h					R-3h			R-0h		R-8h					

Table 23-77. SPIREV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Identification Scheme Used to distinguish different ID schemes. Reads 0x01
29-28	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
27-16	FUNC	R	A05h	Indicates functionally equivalent module family Reads 0xA05
15-11	RTL	R	0h	RTL version number Read value will provide an approximate RTL revision number. The design release version can be obtained from the device specification
10-8	MAJOR	R	3h	Major Revision number Reads 0x3
7-6	CUSTOM	R	0h	Indicates device specific implementation Reads 0x0
5-0	MINOR	R	8h	Minor Revision number Reads 0x8

23.3.2 MSS_MIBSPIB Registers

Table 23-78 lists the memory-mapped registers for the MSS_MIBSPIB. All register offset addresses not listed in Table 23-78 should be considered as reserved locations and the register contents should not be modified.

Table 23-78. MSS_MIBSPIB Registers

Offset	Acronym	Register Name	Section
0h	SPIGCR0	SPI / MibSPI Global Control Register 0	Section 23.3.2.1
4h	SPIGCR1	SPI / MibSPI Global control register 1	Section 23.3.2.2
8h	SPIINT0	SPI / MibSPI Interrupt Enable Register	Section 23.3.2.3
Ch	SPILVL	SPI / MibSPI Interrupt Level Register	Section 23.3.2.4
10h	SPIFLG	SPI / MibSPI Flag Register	Section 23.3.2.5
14h	SPIPC0	SPI / MibSPI Pin Control Register 0 (SPIPC0) - SPIFUN Note: Duplicate Control Bits for SIMO0 & SOMI0 Bit 24 is not physically implemented. it is a mirror of Bit11. Any write to Bit 24 will be reflected on Bit11 and when Bit 24 & Bit 11 simultaneously written, the value of Bit11 will control the SOMI pin. Read value of Bit 24 always reflects the Bit 11 value. This is true for the Bit 24 & Bit 11 of all of SPIPC0 to SPIPC9 registers. Same is true for SIMO pin with Bit16 & Bit 10 of SPIPC0 to SPIPC9 registers.	Section 23.3.2.6
18h	SPIPC1	SPI / MibSPI Pin Control Register 1 (SPIPC1) - SPIDIR	Section 23.3.2.7
1Ch	SPIPC2	SPI / MibSPI Pin Control Register 2 (SPIPC2) - SPIDIN	Section 23.3.2.8
20h	SPIPC3	SPI / MibSPI Pin Control Register 3 (SPIPC3) - SPIDOUT	Section 23.3.2.9
24h	SPIPC4	SPI / MibSPI Pin Control Register 4 (SPIPC4) - SPIDSET	Section 23.3.2.10
28h	SPIPC5	SPI / MibSPI Pin Control Register 5 (SPIPC5) - SPIDCLR	Section 23.3.2.11
2Ch	SPIPC6	SPI / MibSPI Pin Control Register 6 (SPIPC6) - SPIPDR	Section 23.3.2.12
38h	SPIDAT0	SPI / MibSPI Transmit Data Register 0 Note: Accessibility of SPIDAT0 The SPIDAT0 register is not accessible in Multibuffer Mode of MibSPI. It is only accessible in compatibility mode.	Section 23.3.2.13
3Ch	SPIDAT1	SPI / MibSPI Transmit Data Register 1 When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned.	Section 23.3.2.14
40h	SPIBUF	SPI / MibSPI Receive Buffer Register	Section 23.3.2.15
44h	SPIEMU	SPI / MibSPI Emulation Register Note: All the fields ot SPIEMU register are Read-Only. Read operation on this register under any mode will not have any impact on the status of this or any other registers.	Section 23.3.2.16
48h	SPIDELAY	SPI / MibSPI Delay Register	Section 23.3.2.17
4Ch	SPIDEF	SPI / MibSPI Default Chip select Register	Section 23.3.2.18
50h	SPIFMT0	SPI / MibSPI Data Format Register 0	Section 23.3.2.19
54h	SPIFMT1	SPI / MibSPI Data Format Register 1	Section 23.3.2.20
58h	SPIFMT2	SPI / MibSPI Data Format Register 2	Section 23.3.2.21
5Ch	SPIFMT3	SPI / MibSPI Data Format Register 3	Section 23.3.2.22
60h	TGINTVECT0	SPI Interrupt Vector Register 0 / MibSPI Transfer Group Interrupt Vector Register 0	Section 23.3.2.23
64h	TGINTVECT1	SPI Interrupt Vector Register 1 / MibSPI Transfer Group Interrupt Vector Register 1	Section 23.3.2.24

Table 23-78. MSS_MIBSPIB Registers (continued)

Offset	Acronym	Register Name	Section
68h	SPIPC9	SPI/MibSPI Pin Control Register 9 (SPIPC9) - SPISRSEL	Section 23.3.2.25
6Ch	SPIPMCTRL	SPI/MibSPI Parallel/Modulo Mode Control Register	Section 23.3.2.26
70h	MIBSPIE	MibSPI Enable Register	Section 23.3.2.27
74h	TGITENST	MibSPI Transfer Group Interrupt Enable Set Register	Section 23.3.2.28
78h	TGITENCR	MibSPI Transfer Group Interrupt Enable Clear Register	Section 23.3.2.29
7Ch	TGITLVST	MibSPI Transfer Group Interrupt Level Set Register	Section 23.3.2.30
80h	TGITLVCR	MibSPI Transfer Group Interrupt Level Clear Register	Section 23.3.2.31
84h	TGINTFLAG	Transfer Group Interrupt Flag Register	Section 23.3.2.32
90h	TICKCNT	Tick Count Register	Section 23.3.2.33
94h	LTGPEND	Last Transfer Group End Pointer	Section 23.3.2.34
98h	TG0CTRL	MibSPI Transfer Group Control Register The number of transfer groups is scalable by design up to a maximum of 16. Depending on the implementation the number of transfer groups and hence the number of transfer group control register may vary. Each transfer group can be configured via one dedicated control register. The register description below shows one exemplary control register(x) which is identical for all transfer groups. E.g. the control register for transfer group 2 is named “TG2CTRL” and is located at address base0+98h+4*2.	Section 23.3.2.35
9Ch	TG1CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.36
A0h	TG2CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.37
A4h	TG3CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.38
A8h	TG4CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.39
ACh	TG5CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.40
B0h	TG6CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.41
B4h	TG7CTRL	MibSPI Transfer Group Control Register	Section 23.3.2.42
D8h	DMA0CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.43
DCh	DMA1CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.44
E0h	DMA2CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.45
E4h	DMA3CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.46
E8h	DMA4CTRL	MibSPI DMA Channel Control Register	Section 23.3.2.47
F8h	ICOUNT0	MibSPI DMAxCOUNT	Section 23.3.2.48
FCh	ICOUNT1	MibSPI DMAxCOUNT	Section 23.3.2.49
100h	ICOUNT2	MibSPI DMAxCOUNT	Section 23.3.2.50
104h	ICOUNT3	MibSPI DMAxCOUNT	Section 23.3.2.51
108h	ICOUNT4	MibSPI DMAxCOUNT	Section 23.3.2.52
118h	DMACNTLEN	DMA LARGE COUNT register	Section 23.3.2.53
120h	PAR_ECC_CTRL	Parity/ECC Control Register	Section 23.3.2.54
124h	PAR_ECC_STAT	Parity/ECC Status Register	Section 23.3.2.55
128h	UERRADDR1	Uncorrectable Parity or double bit ECC error Address Register - RXRAM	Section 23.3.2.56
12Ch	UERRADDR0	Uncorrectable Parity or double bit ECC error address register - TXRAM	Section 23.3.2.57
130h	RXOVRN_BUF_ADDR	Receive RAM Overrun Buffer Address Register	Section 23.3.2.58

Table 23-78. MSS_MIBSPIB Registers (continued)

Offset	Acronym	Register Name	Section
134h	IOLPBKTSTCR	SPI/MibSPI IO Loopback Test Control Register This register controls test mode for I/O pins. It also controls whether loop-back should be digital or analog ones in this test mode. In addition it contains control bits to induce some of the error condition into the module. These are to be used for test purpose only. All the control/status bits in this register are valid only when IO LPBK TST ENA field is set to “1010”.	Section 23.3.2.59
138h	EXTENDED_PRESCALE1	SPI/MibSPI Extended Prescale Register 1 (EXTENDED_PRESCALE1 for SPIFMT0 and SPIFMT1) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This is an extension of SPIFMT0 and SPIFMT1 registers. For example, EPRESCALE_FMT1(7:0) of EXTENDED_PRESCALE1 and PRESCALE1(7:0) of SPIFMT1 register will always reflect the same contents. Similarly EPRESCALE_FMT0(7:0) and PRESCALE0(7:0) of SPIFMT0 reflect the same contents.	Section 23.3.2.60
13Ch	EXTENDED_PRESCALE2	SPI/MibSPI Extended Prescale Register 2 (EXTENDED_PRESCALE2 for SPIFMT2 and SPIFMT3) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This register is an extension of SPIFMT2 and SPIFMT3 registers. For example, EPRESCALE_FMT2(7:0) of EXTENDED_PRESCALE2 and PRESCALE2(7:0) of SPIFMT2 register will always reflect the same contents. Similarly EPRESCALE_FMT3(7:0) and PRESCALE3(7:0) of SPIFMT3 reflect the same contents.	Section 23.3.2.61
140h	ECCDIAG_CTRL	ECC Diagnostic Control register	Section 23.3.2.62
144h	ECCDIAG_STAT	ECC Diagnostic Status register	Section 23.3.2.63
148h	SBERRADDR1	Single Bit Error Address Register - RXRAM	Section 23.3.2.64
14Ch	SBERRADDR0	Single Bit ECC Error Address Register - TXRAM	Section 23.3.2.65
1FCh	SPIREV	SPI / MibSPI Revision ID Register	Section 23.3.2.66

Complex bit access types are encoded to fit into small table cells. Table 23-79 shows the codes that are used for access types in this section.

Table 23-79. MSS_MIBSPIB Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	-0 R	Returns 0s Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

23.3.2.1 SPIGCR0 Register (Offset = 0h) [reset = 0h]

SPIGCR0 is shown in Figure 23-98 and described in Table 23-80.

Return to Summary Table.

SPI / MibSPI Global Control Register 0

Figure 23-98. SPIGCR0 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							nRESET
R-0h							R/W-0h

Table 23-80. SPIGCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	Reserved, Reads return ‘0’ and writes have no effect.
0	nRESET	R/W	0h	This is the local reset control for the module. This bit needs to be set to ‘1’ before any operation on SPI / MibSPI can be done. Only after setting this bit to ‘1’, the Auto Initialization of Multibuffer RAM starts. Clearing this bit to ‘0’ will result in all of the control and status register values to return to their default values. 0 = SPI / MibSPI is in reset state 1 = SPI / MibSPI is out of reset state.

23.3.2.2 SPIGCR1 Register (Offset = 4h) [reset = 0h]

SPIGCR1 is shown in Figure 23-99 and described in Table 23-81.

Return to Summary Table.

SPI / MibSPI Global control register 1

Figure 23-99. SPIGCR1 Register

31	30	29	28	27	26	25	24
NU4							SPIEN
R-0h							R/W-0h
23	22	21	20	19	18	17	16
NU3							LOOPBACK
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU2							POWERDOWN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NU1						CLKMOD	MASTER
R-0h						R/W-0h	R/W-0h

Table 23-81. SPIGCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	SPIEN	R/W	0h	SPI enable. This bit enables the SPI/MibSPI transfers. This bit must be set to 1 after all other SPI / MibSPI configuration bits have been written. When SPIEN bit is 0 or cleared to 0, the following SPI/MibSPI registers get forced to their default states (to ‘0’s except for RXEMPTY bit in SPIBUF): - Both TX & RX Shift Registers - The TXDATA fields of SPIDAT0 and SPIDAT1 registers - All the fields of SPIFLG register - Contents of SPIBUF & the internal RXBUF registers 0=SPI / MibSPI is not activated for transfers. 1=Activates SPI / MibSPI
23-17	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
16	LOOPBACK	R/W	0h	LOOP BACK. Internal loop-back test mode. The internal self-test option can be enabled by setting this bit. If the SPISIMO and SPISOMI pins are configured with SPI functionality, then the SPISIMO pin is internally connected to the SPISOMI pin. The transmit data is looped back as receive data and is stored in the receive field of the concerned buffer. Externally, during loop-back operation, the SPICLK pin outputs an inactive value and SPISOMI remains in high-impedance state. The SPI / MibSPI has to be initialized in master mode before the loop-back can be selected. If the SPI / MibSPI is initialized in slave mode or a data transfer is ongoing, errors may result. 1 =Internal loop-back test mode enabled. 0 =Internal loop-back test mode disabled. This loopback mode can be used only in Master mode. This automatically selects digital loopback path. When this Loopback mode is selected, CLKMOD bit should be set to ‘1’, meaning that SPICLK can only be internal.
15-9	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
8	POWERDOWN	R/W	0h	POWERDOWN. When active, the SPI / MibSPI state machines enter a powerdown state. 0=MibSPI in active mode 1=MibSPI in powerdown mode
7-2	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
1	CLKMOD	R/W	0h	CLKMOD. Clock mode Selects either an internal or external clock source. This bit also determines the I/O direction of the SPIENA and SPISCS[7:0] pins in functional mode. 0=Clock is external 1=Clock is internal

Table 23-81. SPIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	MASTER	R/W	0h	MASTER: SPISIMO/SPISOMI pin direction determination. Determines the direction of the SPISIMO and SPISOMI pins. This bit determines whether the SPI/MibSPI is in Master mode or Slave mode. This bit also controls the Master-only features like the C2T/T2C counters, C2E/T2E timers, most of the Error conditions specific to Master mode. 0=SPISIMO pin an input, SPISOMI pin an output 1=SPISOMI pin an input, SPISIMO pin an output Note: Although there are two different bits which control the Master/Slave mode functions, only two of their combinations are valid. For compatibility reasons both the bits are retained. For Master mode of operation: MASTER = '1', CLKMOD = '1' For Slave mode of operation: MASTER = '0', CLKMOD = '0' Any other combinations of these two bits may not yield any desirable operation of the module.

23.3.2.3 SPIINT0 Register (Offset = 8h) [reset = 0h]

SPIINT0 is shown in Figure 23-100 and described in Table 23-82.

Return to Summary Table.

SPI / MibSPI Interrupt Enable Register

Figure 23-100. SPIINT0 Register

31	30	29	28	27	26	25	24
NU5							ENABLEHIGHZ
R-0h							R/W-0h
23	22	21	20	19	18	17	16
NU4							DMAREQEN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU3						TXINTENA	RXINTENA
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
NU2	OVRNINTENA	NU1	BITERRENA	DESYNCENA	PARERRENA	TIMEOUTENA	DLENERRENA
R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-82. SPIINT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU5	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	ENABLEHIGHZ	R/W	0h	SPIENA pin high-z enable. When active, the SPIENA pin (when it is configured as a WAIT functional output signal in a slave SPI) is forced to place it is output in high-z when not driving a low signal. If inactive, then the pin will output both a high and a low signal. 0=SPIENA pin is pulled high when not active. 1=SPIENA pin remains in high-z when not active.
23-17	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
16	DMAREQEN	R/W	0h	DMA request enable. Enables the DMA request signal to be generated for both receive and transmit channels. Enable DMA REQ only after setting the SPIEN bit to ‘1’. 0=DMA is not used 1=DMA Requests will be generated. A DMA request will be generated on TX DMA REQ line each time a transmit data is copied to the Shift Register either from TXBUF or directly from SPIDAT0/SPIDAT1 writes. A DMA request will be generated on RX DMA REQ line each time a received data is copied to SPIBUF register either from RXBUF or directly from the Shift Register.
15-10	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
9	TXINTENA	R/W	0h	An interrupt is to be generated everytime data is written to the “Shift Register”, so that a new data can be written to TXBUF. Setting this bit will generate an interrupt if TXINTFLG bit (SPIFLG.9) is set to ‘1’. 0=No interrupt will be generated upon TXINTFLG getting set to ‘1’. 1=Interrupt will be generated upon TXINTFLG getting set to ‘1’. An interrupt request will be generated as soon as this bit is set to ‘1’. By default it will be generated on INT0 line. SPILVL register can be programmed before-hand to change this default.
8	RXINTENA	R/W	0h	An interrupt is to be generated when the RXINTFLAG bit (SPIFLG.8) is set by hardware. Otherwise, no interrupt will be generated. 0=Interrupt will not be generated 1=Interrupt will be generated Both Transmitter Empty & Receiver Full interrupts are valid in SPI or Compatibility mode of MibSPI only. In Multibuffered mode these interrupts will not be generated even if enabled.
7	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.

Table 23-82. SPIINT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	OVRNINTENA	R/W	0h	Overrun interrupt enable. An interrupt is to be generated when the RCVR OVRN flag bit (SPIFLG.6) is set by hardware. Otherwise, no interrupt will be generated. 0=Overrun interrupt will not be generated 1=Overrun interrupt will be generated
5	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
4	BITERRENA	R/W	0h	Enables interrupt on bit error. 1 =Enables an interrupt on a bit error (BITERR = 1). 0 =No interrupt asserted upon bit error.
3	DESYNCENA	R/W	0h	Enables interrupt on de-synchronized slave. DESYNCENA is used in master mode only. 1 =Enables an interrupt on de-synchronization of the slave (DESYNC = 1). 0 =No interrupt asserted upon de-synchronization error.
2	PARERRENA	R/W	0h	Enables interrupt on parity error. 1 =Enables an interrupt on a parity error (PARITYERR = 1). 0 =No interrupt asserted upon parity error.
1	TIMEOUTENA	R/W	0h	Enables interrupt on ENA signal time-out. 1 =Enables an interrupt on a time-out of the ENA signal (TIMEOUT = 1). 0 =No interrupt asserted upon ENA signal time-out.
0	DLENERRENA	R/W	0h	Data Length Error interrupt Enable. 1 = Enables an interrupt when Data Length Error occurs. 0 = No interrupt is generated upon Data Length Error. A Data Length Error occurs under the following conditions. Master: In a 4-pin with SPIENA mode or 5-pin mode, if the SPIENA pin from the slave is deasserted before the Master has completed its transfer, the Data Length Error is set. That is, if the Character Length counter has not completed counting while SPIENA deassertion is detected, then it means that the Slave has neither received full data from the Master nor has it transmitted complete data. Slave: In a 4-pin with ChipSelects mode or 5-pin mode, if the incoming valid SPISCS pin is de-activated before the Character Length counter completes counting, then Data Length Error is set.

23.3.2.4 SPILVL Register (Offset = Ch) [reset = 0h]

SPILVL is shown in Figure 23-101 and described in Table 23-83.

Return to Summary Table.

SPI / MibSPI Interrupt Level Register

Figure 23-101. SPILVL Register

31	30	29	28	27	26	25	24
NU3							
R-0h							
23	22	21	20	19	18	17	16
NU3							
R-0h							
15	14	13	12	11	10	9	8
NU3						TXINTLVL	RXINTLVL
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
NU2	OVRNINTLVL	NU1	BITERRLVL	DESYNCLVL	PARERRLVL	TIMEOUTLVL	DLENERRLVL
R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 23-83. SPILVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
9	TXINTLVL	R/W	0h	Transmit Interrupt Level. 1 =Transmit interrupt is mapped to interrupt line INT1. 0 =Transmit interrupt is mapped to interrupt line INT0.
8	RXINTLVL	R/W	0h	Receive interrupt level. 1 =Receive interrupt is mapped to interrupt line INT1. 0 =Receive interrupt is mapped to interrupt line INT0.
7	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
6	OVRNINTLVL	R/W	0h	Receive Overrun interrupt level. 1 =Receive Overrun interrupt is mapped to interrupt line INT1. 0 =Receive Overrun interrupt is mapped to interrupt line INT0.
5	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
4	BITERRLVL	R/W	0h	Bit error interrupt level. 1 =bit error interrupt is mapped to interrupt line INT1. 0 =bit error interrupt is mapped to interrupt line INT0.
3	DESYNCLVL	R/W	0h	De-synchronized slave interrupt level. DESYNCLVL is used in master mode only. 1 =An interrupt due to de-synchronization of the slave (DESYNC = 1) is mapped to interrupt line INT1. 0 =An interrupt due to de-synchronization of the slave (DESYNC = 1) is mapped to interrupt line INT0.
2	PARERRLVL	R/W	0h	Parity error interrupt level. 1 =A parity error interrupt (PARITYERR = 1) is mapped to interrupt line INT1. 0 =A parity error interrupt (PARITYERR = 1) is mapped to interrupt line INT0.
1	TIMEOUTLVL	R/W	0h	SPIENA pin Time-out interrupt level. 1 =An interrupt on a time-out of the ENA signal (TIMEOUT = 1) is mapped to interrupt line INT1. 0 =An interrupt on a time-out of the ENA signal (TIMEOUT = 1) is mapped to interrupt line INT0.
0	DLENERRLVL	R/W	0h	Data Length Error interrupt Enable Level. 1 = An interrupt on Data Length Error is mapped to interrupt line INT1. 0 = An interrupt on Data Length Error is mapped to interrupt line INT0.

23.3.2.5 SPIFLG Register (Offset = 10h) [reset = 0h]

SPIFLG is shown in Figure 23-102 and described in Table 23-84.

Return to Summary Table.

SPI / MibSPI Flag Register

Figure 23-102. SPIFLG Register

31	30	29	28	27	26	25	24
NU4							BUFINITACTIVE
R-0h							R-0h
23	22	21	20	19	18	17	16
NU3							
R-0h							
15	14	13	12	11	10	9	8
NU3						TXINTFLG	RXINTFLG
R-0h						R-0h	0h
7	6	5	4	3	2	1	0
NU2	OVRNINTFLG	NU1	BITERRFLG	DESYNCFLG	PARERRFLG	TIMEOUTFLG	DLENERRFLG
R-0h	0h	R-0h	0h	0h	0h	0h	0h

Table 23-84. SPIFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	BUFINITACTIVE	R	0h	Indicates the status of Multibuffer initialization process. Software should poll this bit to determine if it can proceed with the configuration of Multibuffer mode registers or Multibuffer RAM handling. Refer to Section 3.10.7 for details on Initialization of Multibuffer RAM. 1 = Multibuffer RAM is still being initialized. Do not attempt to write to either Multibuffer RAM or any Multibuffer mode registers. Refer to Section 3.3 for a classification of registers into compatibility mode and Multibuffer mode. 0 = Multibuffer RAM initialization is complete. This bit will show a value of ‘1’ as long as the nRESET bit is ‘0’, but does not really indicate that Buffer initialization is underway. Internal automatic buffer initialization starts only when the nRESET bit is set to ‘1’. For SPI, this bit reads ‘1’ always. For MibSPI, BUF INIT ACTIVE bit will show up as ‘1’ for a maximum of 128/256 (will vary depending upon the actual size of the Multibuffer RAM implemented) VCLK cycles after the nRESET bit in GCR0 is set to ‘1’ and then settle to ‘0’. If Auto Memory Initialization is triggered through System (MEM_AUTO_INIT pulse), then BUF INIT ACTIVE bit will show up as ‘1’ for a maximum of 128/256 (will vary depending upon the actual size of the Multibuffer RAM implemented) VCLK cycles and then settle to ‘0’.
23-10	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
9	TXINTFLG	R	0h	Transmitter Empty Interrupt Flag. Serves as an interrupt flag indicating that Transmit Buffer (TXBUF) is empty and a new data can be written to it. This flag is set when a data is copied to the “Shift Register” either directly or from the TXBUF register. This bit is cleared by one of following ways: Writing a new data to either SPIDAT0 or SPIDAT1 Writing a ‘0’ to SPIEN (SPIGCR1.24) 0= Transmit Buffer is now full. No interrupt pending for Transmitter Empty 1= Transmit Buffer is empty. An interrupt is pending to fill the transmitter.

Table 23-84. SPIFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RXINTFLG		0h	Receiver Full Interrupt Flag. This flag is set when a word is received and copied into the buffer register (SPIBUF). If RXINTEN is enabled, an interrupt is also generated. During emulation mode, however, a read to the emulation register (SPIEMU) does not clear this flag bit. This bit is cleared under the following ways: Reading the SPIBUF register Reading TGINTVECT0 or TGINTVECT1 register when there is a “Receive Buffer Full” interrupt Writing a ‘1’ to this bit Writing a ‘0’ to SPIEN (SPIGCR1.24) System reset 0= No new received data pending. Receive buffer is Empty. 1= A newly received data is ready to be read. Receive buffer is full. Note: Exception for clearing of RXINT If both SPIBUF and RXBUF (internal buffer) are full, then, reading TGINTVECT0 or TGINTVECT1 register (while it shows 10010) does not clear the RXINTFLG in SPIFLG register. In this case, only way to clear the Interrupt is to read the SPIBUF (twice) and clear all the received data. Note: Side effects of Write Clear to RXINTFLG Clearing RXINTFLG bit by writing a ‘1’ before reading the SPIBUF sets the RXEMPTY bit of the SPIBUF register too. This way, one can ignore a received data. However, if the internal RXBUF is already full, the data from RXBUF will be copied to SPIBUF and RXEMPTY bit will be cleared again. SPIBUF contents should be read first if this situation needs to be avoided.
7	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
6	OVRNINTFLG		0h	Receiver overrun flag. The SPI / MibSPI hardware sets this bit when a receive operation completes before the previous character has been read from the receive buffer. The bit indicates that the last received character has been overwritten and therefore lost. The SPI / MibSPI will generate an interrupt request if this bit is set and the OVRN INTEN bit (SPIINT0.6) is set high. 0 = Overrun condition did not occur 1 = Overrun condition has occurred In SPI or Compatibility mode of MibSPI, this bit is cleared under the following conditions: Reading TGINTVECT0 or TGINTVECT1 register when there is a “Receive Buffer Overrun” interrupt Writing a ‘1’ to OVRNINTFLG in SPIFLG register itself Reading SPIBUF register does not clear this OVRNINTFLG bit. If an RXOVRN interrupt is detected, then the SPIBUF may need to be read twice to get to the Overrun buffer. This is due to the fact that the Overrun will always occur to the internal RXBUF. Each read to the SPIBUF will result in RXBUF contents (if it is full) getting copied to SPIBUF. Note: A special condition under which OVRNINTFLG flag gets set. If both SPIBUF & RXBUF are already full and while another buffer receive is underway, if any errors like TIMEOUT, BITERR & DLEN_ERR occur, then RXOVR in RXBUF & OVRNINTFLG in SPIFLG registers will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a normal Receiver Overrun. In Multibuffer mode of MibSPI, this bit is cleared under the following conditions. Reading the RXOVRN_BUF_ADDR register Writing a ‘1’ to OVRNINTFLG in SPIFLG register itself In Multibuffer mode, if OVRNINTFLG is set, then the address of the buffer which experienced the Overrun is available in RXOVRN_BUF_ADDR.
5	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
4	BITERRFLG		0h	Mismatch of internal transmit data and transmitted data. 1 =A bit error occurred. The SPI / MibSPI samples the signal of the transmit pin (master: SIMO, slave: SOMI) at the receive point (half clock cycle after transmit point). If the sampled value differs from the transmitted value a bit error is detected and the Flag BITERR is set. If BITERRENA is set an interrupt is asserted. A possible reason for a bit error can be a to high bit rate / capacitive load or another master/slave trying to transmit at the same time. 0 =No bit error occurred. This flag can be cleared by one of the following ways. Write a ‘1’ to this bit. Set SPIEN bit to ‘0’.

Table 23-84. SPIFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	DESYNCFLG		0h	De-synchronization of slave device. De-synchronization monitor is active in master mode only. 1 = A slave device is de-synchronized. The master monitors the ENAble signal coming from the slave device and sets the DESYNC flag after the last bit is transmitted plus tT2EDELAY (see Section 8.21). If DESYNCENA is set an interrupt is asserted. De-synchronization can occur if a slave device misses a clock edge coming from the master. 0 =No slave de-synchronization detected. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'. Note: Inconsistency of Desync flag in SPI/Compatibility mode MibSPI Due to the nature of this Error, under some circumstances it is possible for Desync error detected for the previous buffer to be visible in the current buffer. This is due to the fact that Receive Completion flag/interrupt will be generated when the buffer transfer is completed. But Desync will be detected after the buffer transfer is completed. So, if VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct Desync condition. This inconsistency in Desync flag is valid only in SPI or Compatibility mode of MibSPI. In Multibuffer mode, Desync flag is always guaranteed to be for the current buffer.
2	PARERRFLG		0h	Calculated parity differs from received parity bit. 1 =A parity error occurred. If the parity generator is enabled (can be selected individually for each buffer) an even or odd parity bit is added at the end of a data word (see Section 8.23). During reception of the data word the parity generator calculates the reference parity and compares it to the received parity bit. In the event of a mismatch the PARITYERR flag is set and an interrupt is asserted if PARERRENA is set. 0 =No parity error detected. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'.
1	TIMEOUTFLG		0h	Time-out due to non-activation of ENA signal. 1 =An ENA signal time-out occurred. The SPI / MibSPI generates a time-out because the slave hasn't responded in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition the TIMOUT flag in the status field of the corresponding buffer is set. The transmit request of the concerned buffer is cleared, i.e. the SPI / MibSPI doesn't re-start a data transfer from this buffer. 0 =No ENA-signal time-out occurred. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'.
0	DLENERRFLG		0h	Data Length Error Flag. 1 = A Data Length Error has occured. 0 = No Data Length Error has occured. This flag can be cleared by one of the following ways. Write a '1' to this bit. Set SPIEN bit to '0'. A Data Length Error occurs under the following conditions. Master: In a 4-pin with SPIENA mode or 5-pin mode, if the SPIENA pin from the slave is deasserted before the Master has completed its transfer, the Data Length Error is set. That is, if the Character Length counter has not completed counting while SPIENA pin deassertion is detected, then it means that the Slave has neither received full data from the Master nor has it transmitted complete data. Slave: In a 4-pin with ChipSelects mode or 5-pin mode, if the incoming valid SPISCS pin is de-activated before the Character Length counter completes counting, then Data Length Error is set. Note: Clearing of Transmission Error Flags in SPIBUF during Error conditions Whenever any Transmission Errors (TIMEOUT, BITERR, DLEN_ERR, PARITY_ERR, DESYNC) are detected, and the Error Flag are cleared by writing to the Error bit in SPIFLG register, the corresponding Error flag in SPIBUF does not get cleared. Software needs to read the SPIBUF until it becomes empty before proceeding. This ensures that all the older status bits in SPIBUF are cleared before starting the next transfer.

23.3.2.6 SPIPC0 Register (Offset = 14h) [reset = 0h]

SPIPC0 is shown in Figure 23-103 and described in Table 23-85.

Return to Summary Table.

SPI / MibSPI Pin Control Register 0 (SPIPC0) - SPIFUN Note: Duplicate Control Bits for SIMO0 & SOMI0 Bit 24 is not physically implemented. it is a mirror of Bit11. Any write to Bit 24 will be reflected on Bit11 and when Bit 24 & Bit 11 simultaneously written, the value of Bit11 will control the SOMI pin. Read value of Bit 24 always reflects the Bit 11 value. This is true for the Bit 24 & Bit 11 of all of SPIPC0 to SPIPC9 registers. Same is true for SIMO pin with Bit16 & Bit 10 of SPIPC0 to SPIPC9 registers.

Figure 23-103. SPIPC0 Register

31	30	29	28	27	26	25	24
SOMIFUN							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOFUN							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIFUN0	SIMOFUN0	CLKFUN	ENAFUN
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSFUN							
R/W-0h							

Table 23-85. SPIPC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIFUN	R/W	0h	Slave out, master in function. Determines whether the SPISOMIx pins are to be used as a general-purpose I/O pin or as a SPI / MibSPI functional pin. 0=SPISOMIx pin is a GPIO 1=SPISOMIx pin is a SPI / MibSPI functional pin
23-16	SIMOFUN	R/W	0h	Slave in, master out function. Determines whether the SPISIMOx pin is to be used as a general-purpose I/O pin or as a SPI / MibSPI functional pin. 0=SPISIMOx pin is a GPIO 1=SPISIMOx pin is a SPI / MibSPI functional pin Note: Generic based bit implementation Register bits 31 to 24 and 23 to 16 of SPIPC0 to SPIPC9 are implemented depending upon the generic parameter NUM_PARLL_PINS which determines the number of SIMO/SOMI data lines to be supported. Only if 8 dataline support is selected at the time of logic synthesis, bits 31 to 16 are implemented. Un-implemented bits return ‘0’ upon read and are not writable.
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIFUN0	R/W	0h	Slave out, master in function. Determines whether the SPISOMI0 pin is to be used as a general-purpose I/O pin or as a SPI / MibSPI functional pin. 0=SPISOMI0 pin is a GPIO 1=SPISOMI0 pin is a SPI / MibSPI functional pin Note: Bit 11 or bit 24 can be used to set the function mode for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOFUN0	R/W	0h	Slave in, master out function. Determines whether the SPISIMO0 pin is to be used as a general-purpose I/O pin, or as a SPI / MibSPI functional pin. 0=SPISIMO0 pin is a GPIO 1=SPISIMO0 pin is a SPI / MibSPI functional pin Note: Bit 10 or bit 16 can be used to set the function mode for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKFUN	R/W	0h	SPI / MibSPI clock function. Determines whether the SPICLK pin is to be used as a general-purpose I/O pin, or as a SPI / MibSPI functional pin. 0=SPICLK pin is a GPIO 1=SPICLK pin is a SPI / MibSPI functional pin

Table 23-85. SPIPC0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ENAFUN	R/W	0h	SPIENA function. Determines whether the SPIENA pin is to be used as a general-purpose I/O pin, or as a SPI / MibSPI functional pin. 0=SPIENA pin is a GPIO 1=SPIENA pin is a SPI / MibSPI functional pin
7-0	SCSFUN	R/W	0h	SPISCS[7:0] function. Determines whether the SPISCSx pins are to be used as a general-purpose I/O pins or as SPI functional pins. If the slave SPISCSx pins are in functional mode and receive an inactive high signal, the slave SPI will place it is output in high-z and disable shifting. 0=SPISCSx pin is a GPIO 1=SPISCSx pin is a SPI functional pin Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSFUN[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read ‘0’ always.

23.3.2.7 SPIPC1 Register (Offset = 18h) [reset = 0h]

SPIPC1 is shown in Figure 23-104 and described in Table 23-86.

Return to Summary Table.

SPI / MibSPI Pin Control Register 1 (SPIPC1) - SPIDIR

Figure 23-104. SPIPC1 Register

31	30	29	28	27	26	25	24
SOMIDIR							
R/W-0h							
23	22	21	20	19	18	17	16
SIMODIR							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIDIR0	SIMODIR0	CLKDIR	ENADIR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSDIR							
R/W-0h							

Table 23-86. SPIPC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIDIR	R/W	0h	SPISOMIx direction. Controls the direction of the SPISOMIx pin when it is used as a general-purpose I/O pin. If the SPISOMIx pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISOMIx pin is an input 1=SPISOMIx pin is an output
23-16	SIMODIR	R/W	0h	SPISIMOx direction. Controls the direction of the SPISIMOx pin when it is used as a general-purpose I/O pin. If the SPISIMOx pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISIMOx pin is an input 1=SPISIMOx pin is an output
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIDIR0	R/W	0h	SPISOMI0 direction. Controls the direction of the SPISOMI0 pin when it is used as a general-purpose I/O pin. If the SPISOMI0 pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISOMI0 pin is an input 1=SPISOMI0 pin is an output Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMODIR0	R/W	0h	SPISIMO0 direction. Controls the direction of the SPISIMO0 pin when it is used as a general-purpose I/O pin. If the SPISIMO0 pin is used as a SPI / MibSPI functional pin, the I/O direction is determined by the MASTER bit. 0=SPISIMO0 pin is an input 1=SPISIMO0 pin is an output Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKDIR	R/W	0h	SPICLK direction. Controls the direction of the SPICLK pin when it is used as a general-purpose I/O pin. In functional mode, the I/O direction is determined by the CLKMOD bit. 0=SPICLK pin is an input 1=SPICLK pin is an output
8	ENADIR	R/W	0h	SPIENA direction. Controls the direction of the SPIENA pin when it is used as a general-purpose I/O. If the SPIENA pin is used as a functional pin, then the I/O direction is determined by the CLKMOD bit (SPIGCR1.1). 0=SPIENA pin is an input 1=SPIENA pin is an output

Table 23-86. SPIPC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	SCSDIR	R/W	0h	SPISCS[7:0] direction. Controls the direction of the SPISCSx pins when they are used as a general-purpose I/O pin. Each pins could be configured independently from the others If the SPISCSx is used as a SPI functional pin, the I/O direction is determined by the CLKMOD bit (SPIGCR1.1). 0=SPISCSx pin is an input 1=SPISCSx pin is an output Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSDIR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read ‘0’ always.

23.3.2.8 SPIPC2 Register (Offset = 1Ch) [reset = 0h]

SPIPC2 is shown in Figure 23-105 and described in Table 23-87.

Return to Summary Table.

SPI / MibSPI Pin Control Register 2 (SPIPC2) - SPIDIN

Figure 23-105. SPIPC2 Register

31	30	29	28	27	26	25	24
SOMIDIN							
R-0h							
23	22	21	20	19	18	17	16
SIMODIN							
R-0h							
15	14	13	12	11	10	9	8
NU				SOMIDIN0	SIMODIN0	CLKDIN	ENADIN
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SCSDIN							
R-0h							

Table 23-87. SPIPC2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIDIN	R	0h	SPISOMIx data in. Reflects the value of the SPISOMIx pin. 0=Current value on SPISOMIx pin is logic 0. 1=Current value on SPISOMIx pin is logic 1
23-16	SIMODIN	R	0h	SPISIMOx data in. Reflects the value of the SPISIMOx pin. 0=Current value on SPISIMOx pin is logic 0. 1=Current value on SPISIMOx pin is logic 1
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIDIN0	R	0h	SPISOMI0 data in. Reflects the value of the SPISOMI0 pin. 0=Current value on SPISOMI0 pin is logic 0. 1=Current value on SPISOMI0 pin is logic 1 Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMODIN0	R	0h	SPISIMO0 data in. Reflects the value of the SPISIMO0 pin. 0=Current value on SPISIMO0 pin is logic 0. 1=Current value on SPISIMO0 pin is logic 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKDIN	R	0h	Clock data in. Reflects the value of the SPICLK pin. 0=Current value on SPICLK pin is logic 0. 1=Current value on SPICLK pin is logic 1
8	ENADIN	R	0h	SPIENA data in. Reflects the value of the SPIENA pin. 0=Current value on SPIENA pin is logic 0. 1=Current value on SPIENA pin is logic 1
7-0	SCSDIN	R	0h	SPISCS[7:0] data in. Reflects the value of the SPISCSx pins. 0=Current value on SPISCSx pin is logic 0. 1=Current value on SPISCSx pin is logic 1 Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSDIN[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will read ‘0’ always.

23.3.2.9 SPIPC3 Register (Offset = 20h) [reset = 0h]

SPIPC3 is shown in Figure 23-106 and described in Table 23-88.

Return to Summary Table.

SPI / MibSPI Pin Control Register 3 (SPIPC3) - SPIDOUT

Figure 23-106. SPIPC3 Register

31	30	29	28	27	26	25	24
SOMIDOUT							
R/W-0h							
23	22	21	20	19	18	17	16
SIMODOUT							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIDOUT0	SIMODOUT0	CLKDOUT	ENADOUT
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSDOUT							
R/W-0h							

Table 23-88. SPIPC3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIDOUT	R/W	0h	SPISOMIx dataout write. Only active when the SPISOMIx pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISOMIx pin is logic 0. 1=Current value on SPISOMIx pin is logic 1
23-16	SIMODOUT	R/W	0h	SPISIMOx dataout write. Only active when the SPISIMOx pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISIMOx pin is logic 0. 1=Current value on SPISIMOx pin is logic 1
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIDOUT0	R/W	0h	SPISOMI0 dataout write. Only active when the SPISOMI0 pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISOMI0 pin is logic 0. 1=Current value on SPISOMI0 pin is logic 1. Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMODOUT0	R/W	0h	SPISIMO0 dataout write. Only active when the SPISIMO0 pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPISIMO0 pin is logic 0. 1=Current value on SPISIMO0 pin is logic 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKDOUT	R/W	0h	SPICLK dataout write. Only active when the SPICLK pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPICLK pin is logic 0. 1=Current value on SPICLK pin is logic 1
8	ENADOUT	R/W	0h	SPIENA dataout write. Only active when the SPIENA pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. 0=Current value on SPIENA pin is logic 0. 1=Current value on SPIENA pin is logic 1

Table 23-88. SPIPC3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	SCSDOUT	R/W	0h	SPISCS[7:0] dataout write. Only active when the SPISCSx pins are configured as a general-purpose I/O pins and configured as an output pins. The value of these bit indicates the value sent to the pins. 0=Current value on SPISCSx pin is logic 0. 1=Current value on SPISCSx pin is logic 1 Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSDOUT[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always.

23.3.2.10 SPIPC4 Register (Offset = 24h) [reset = 0h]

SPIPC4 is shown in Figure 23-107 and described in Table 23-89.

Return to Summary Table.

SPI / MibSPI Pin Control Register 4 (SPIPC4) - SPIDSET

Figure 23-107. SPIPC4 Register

31	30	29	28	27	26	25	24
SOMISET							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOSET							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMISET0	SIMOSET0	CLKSET	ENASET
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSSET							
R/W-0h							

Table 23-89. SPIPC4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMISET	R/W	0h	SPISOMIx dataout set. Only active when the SPISOMIx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISOMIDOUTx bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISOMIx pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUTx is 0. 1= Current value on SOMIDOUTx is 1.
23-16	SIMOSET	R/W	0h	SPISIMOx dataout set. Only active when the SPISIMOx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISIMODOUTx bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISIMOx pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUTx is 0. 1= Current value on SIMODOUTx is 1
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMISET0	R/W	0h	SPISOMI0 dataout set. Only active when the SPISOMI0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISOMIDOUT bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISOMI0 pin if it is in General Purpose O/P mode Read: 0= Current value on SOMIDOUT0 is 0. 1= Current value on SOMIDOUT0 is 1. Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOSET0	R/W	0h	SPISIMO0 dataout set. Only active when the SPISIMO0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding SPISIMODOUT bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPISIMO0 pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUT0 is 0. 1= Current value on SIMODOUT0 is 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKSET	R/W	0h	SPICLK dataout set. Only active when the SPICLK pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit sets the corresponding CLKDOUT bit to ‘1’. Write: 0= Has no effect 1= Logic 1 placed on SPICLK pin if it is in General Purpose O/P mode Read: 0= Current value on CLKDOUT pin is logic 0. 1= Current value on CLKDOUT pin is logic 1

Table 23-89. SPIPC4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ENASET	R/W	0h	SPIENA dataout set. Only active when the SPIENA pin is configured as a general-purpose output pin. A value of '1' written to this bit sets the corresponding ENABLEDOUT bit to '1'. Write: 0= Has no effect 1= Logic 1 placed on SPIENA pin if it is in General Purpose O/P mode Read: 0= Current value on ENADOUT is 0. 1= Current value on ENADOUT is 1
7-0	SCSSET	R/W	0h	SPISCS[7:0] dataout set. Only active when the SPISCSx pins are configured as a general-purpose output pins. A value of '1' written to these bits set the corresponding SCSDOUT bit to '1'. Write: 0= Has no effect 1= Logic 1 placed on SPISCSx pin if it is in General Purpose O/P mode Read: 0= Current value on SCSDOUTx is 0. 1= Current value on SCSDOUTx is 1. Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSSET[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always. Note: Register Read Read of SPIPC4 register gives out contents of the SPIPC3 register.

23.3.2.11 SPIPC5 Register (Offset = 28h) [reset = 0h]

SPIPC5 is shown in Figure 23-108 and described in Table 23-90.

Return to Summary Table.

SPI / MibSPI Pin Control Register 5 (SPIPC5) - SPIDCLR

Figure 23-108. SPIPC5 Register

31	30	29	28	27	26	25	24
SOMICLR							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOCLR							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMICLR0	SIMOCLR0	CLKCLR	ENACLR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSCLR							
R/W-0h							

Table 23-90. SPIPC5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMICLR	R/W	0h	SPISOMIx dataout clear. Only active when the SPISOMIx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SPISOMIDOUTx bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISOMIx pin if it is in General Purpose O/P mode Read: 0= Current value on SOMIDOUTx is 0. 1= Current value on SOMIDOUTx is 1.
23-16	SIMOCLR	R/W	0h	SPISIMOx dataout clear. Only active when the SPISIMOx pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SPISIMODOUTx bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISIMOx pin if it is in General Purpose O/P mode Read: 0=Current value on SIMODOUTx is 0. 1=Current value on SIMODOUTx is 1.
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMICLR0	R/W	0h	SPISOMI0 dataout clear. Only active when the SPISOMI0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SPISOMIDOUT bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISOMI0 pin if it is in General Purpose O/P mode Read: 0= Current value on SOMIDOUT0 is 0. 1= Current value on SOMIDOUT0 is 1. Note: Bit 11 or bit 24 can be used to set the direction for pin SOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOCLR0	R/W	0h	SPISIMO0 dataout clear. Only active when the SPISIMO0 pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding SIMODOUT0 bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPISIMO0 pin if it is in General Purpose O/P mode Read: 0= Current value on SIMODOUT0 is 0. 1= Current value on SIMODOUT0 is 1. Note: Bit 10 or bit 16 can be used to set the direction for pin SIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKCLR	R/W	0h	SPICLK dataout clear. Only active when the SPICLK pin is configured as a general-purpose output pin. A value of ‘1’ written to this bit clears the corresponding CLKDOUT bit to ‘0’. Write: 0= Has no effect 1= Logic 0 placed on SPICLK pin if it is in General Purpose O/P mode Read: 0= Current value on CLKDOUT is 0. 1= Current value on CLKDOUT is 1.

Table 23-90. SPIPC5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ENACLR	R/W	0h	SPIENA dataout clear. Only active when the SPIENA pin is configured as a general-purpose output pin. A value of '1' written to this bit clears the corresponding ENABLEDOUT bit to '0'. Write: 0= Has no effect 1= Logic 0 placed on SPIENA pin if it is in General Purpose O/P mode Read: 0= Current value on ENADOUT is 0. 1= Current value on ENADOUT is 1.
7-0	SCSCLR	R/W	0h	SPISCS[7:0] dataout clear. Only active when the SPISCSx pins are configured as a general-purpose output pins. A value of '1' written to this bit clears the corresponding SCSDOUT bit to '0.. Write: 0= Has no effect 1= Logic 0 placed on SPISCSx pin if it is in General Purpose O/P mode Read: 0= Current value on SCSDOUTx is 0. 1= Current value on SCSDOUTx is 1 Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSCLR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always. Note: Register Read Read of SPIPC5 register gives out contents of the SPIPC3 register.

23.3.2.12 SPIPC6 Register (Offset = 2Ch) [reset = 0h]

SPIPC6 is shown in Figure 23-109 and described in Table 23-91.

Return to Summary Table.

SPI / MibSPI Pin Control Register 6 (SPIPC6) - SPIPDR

Figure 23-109. SPIPC6 Register

31	30	29	28	27	26	25	24
SOMIPDR							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOPDR							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMIPDR0	SIMOPDR0	CLKPDR	ENAPDR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSPDR							
R/W-0h							

Table 23-91. SPIPC6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMIPDR	R/W	0h	SPISOMIx Open drain enable Enables Open drain capability for the pin SOMIx if the following conditions are met. SOMIDIRx = 1 (SPISOMI0 pin configured in GPIO mode as output pin) SOMIDOUTx = 1 0 = Output value on SPISOMIx pin is logic ‘1’ 1 = Output pin SPISOMIx is Tri-stated
23-16	SIMOPDR	R/W	0h	SPISIMOx Open drain enable Enables Open drain capability for the pin SPISIMOx if the following conditions are met. SIMODIRx = 1 (SPISIMOx pin configured in GPIO mode as output pin) SIMODOUTx = 1 0 = Output value on SPISIMOx pin is logic ‘1’ 1 = Output pin SPISIMOx is Tri-stated
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMIPDR0	R/W	0h	SPISOMI0 Open drain enable Enables Open drain capability for the pin SPISOMI if the following conditions are met. SOMIDIR0 = 1 (SPISOMI0 pin configured in GPIO mode as output pin) SOMIDOUT0 = 1 0 = Output value on SPISOMI0 pin is logic ‘1’ 1 = Output pin SPISOMI0 is Tri-stated Note: Bit 11 or bit 24 can be used to set the direction for pin SPISOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOPDR0	R/W	0h	SPISIMO0 Open drain enable Enables Open drain capability for the pin SPISIMO0 if the following conditions are met. SIMODIR0 = 1 (SPISIMO pin configured in GPIO mode as output pin) SIMODOUT0 = 1 0 = Output value on SPISIMO0 pin is logic ‘1’ 1 = Output pin SPISIMO0 is Tri-stated Note: Bit 10 or bit 16 can be used to set the direction for pin SPISIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKPDR	R/W	0h	SPICLK Open drain enable Enables Open drain capability for the pin CLK if the following conditions are met. CLKDIR = 1 (SPICLK pin configured in GPIO mode as output pin) CLKDOUT = 1 0 = Output value on SPICLK pin is logic ‘1’ 1 = Output pin SPICLK is Tri-stated
8	ENAPDR	R/W	0h	SPIENA Open drain enable Enables Open drain capability for the pin SPIENA if the following conditions are met. ENABLEDIR = 1 (SPIENA pin configured in GPIO mode as output pin) ENABLEDOUT = 1 0 = Output value on SPIENA pin is logic ‘1’ 1 = Output pin SPIENA is Tri-stated

Table 23-91. SPIPC6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	SCSPDR	R/W	0h	SPISCSx Open drain enable Enables Open drain capability for the pin SPISCSx if the following conditions are met. SCSDIRx = 1 (SPISCS pin configured in GPIO mode as output pin) SCSDOUTx = 1 0 = Output value on SPISCSx pin is logic '1' 1 = Output pin SPISCSx is Tri-stated Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSPDR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always.

23.3.2.13 SPIDAT0 Register (Offset = 38h) [reset = 0h]

SPIDAT0 is shown in Figure 23-110 and described in Table 23-92.

Return to Summary Table.

SPI / MibSPI Transmit Data Register 0 Note: Accessibility of SPIDAT0 The SPIDAT0 register is not accessible in Multibuffer Mode of MibSPI. It is only accessible in compatibility mode.

Figure 23-110. SPIDAT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																TXDATA															
R-0h																R/W-0h															

Table 23-92. SPIDAT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
15-0	TXDATA	R/W	0h	SPI / MibSPI Transmit Data. When written, these bits will be copied to the Shift Register if it is empty. If the Shift Register is not empty, the TXBUF will hold the written values. SPIEN (SPICGR1.24) must be set to 1 before this register can be written to. Writing a 0 to the SPIEN register forces the lower 16 bits of the SPIDAT0 register to 0x00. When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned. As the data is shifted out from either the MSB or the LSB of Transmit Shift Register depending upon SHIFTDIR bit (SPIFMTx.20). Simultaneously, the received bit will be shifted into the Receive Shift Register either through LSB or MSB depending upon SHIFTDIR bit. This allows the concurrent transmission and reception of data. Note: Irrespective of the character length, the Transmit data should be right justified before writing to SPIDAT0 register. The default Data Format Control register for SPIDAT0 is SPIFMT0. However it is possible to reprogram the DFSEL[1:0] fields of SPIDAT1 prior to using SPIDAT0, to select a different SPIFMTx register.

23.3.2.14 SPIDAT1 Register (Offset = 3Ch) [reset = 0h]

SPIDAT1 is shown in Figure 23-111 and described in Table 23-93.

Return to Summary Table.

SPI / MibSPI Transmit Data Register 1 When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned.

Figure 23-111. SPIDAT1 Register

31	30	29	28	27	26	25	24
NU2			CSHOLD	NU1	WDEL	DFSEL	
R-0h			R/W-0h	R-0h	R/W-0h	R/W-0h	
23	22	21	20	19	18	17	16
CSNR							
R/W-0h							
15	14	13	12	11	10	9	8
TXDATA							
R/W-0h							
7	6	5	4	3	2	1	0
TXDATA							
R/W-0h							

Table 23-93. SPIDAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
28	CSHOLD	R/W	0h	Chip select hold mode. In SPI or compatibility mode MibSPI, the CSHOLD bit is supported in master mode only. In slave mode, this bit is ignored. CSHOLD defines the behavior of the chip select line at the end of a data transfer. 1 =The chip select signal is held active at the end of a transfer until a control field with new data and control information is loaded into SPIDAT1. If the new chip select information equals the previous one, the active chip select signal is extended until the end of transfer with CSHOLD cleared or until the chip select information changes. 0 =The chip select signal is deactivated at the end of a transfer after the T2CDELAY time has passed. If two consecutive transfers are dedicated to the same chip select this chip select signal will be deactivated for atleast 2VCLK cycles before it is activated again.
27	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
26	WDEL	R/W	0h	Enable the delay counter at the end of the current transaction. WDELAY bit is supported in Master mode only. In Slave mode, this bit will be ignored. 1 = After a transaction, WDELAY of the corresponding data format will be loaded into the delay counter. No transaction will be performed until the WDELAY counter overflows. The SPISCS pins will be de-activated for atleast (WDELAY + 2) * VCLK_Period duration. 0 = No delay will be inserted. However, SPISCS pins will still be de-activated for atleast for 2VCLK cycles if CSHOLD = ‘0’. In SPI or Compatibility mode of MibSPI, the duration for which the SPISCS pin remaining de-activated will also depend upon time taken to supply a new data after completing the shifting operation. If the internal buffer - TXBUF is already full, then the SPISCS will be deasserted for atleast 2 VCLK cycles(if WDEL = ‘0’).
25-24	DFSEL	R/W	0h	DFSEL1 DFSEL0 Description 0 0 Data word format 0 is selected 0 1 Data word format 1 is selected 1 0 Data word format 2 is selected 1 1 Data word format 3 is selected

Table 23-93. SPIDAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-16	CSNR	R/W	0h	Chip select number. CSNR defines the chip select that shall be activated during the data transfer. The value of CSNR[7:0] will be driven on SPISCS[7:0] lines during the transfer. Note: Effect of NUM_CS_PINS generic on CSNR bits. Actual number of bits implemented in CSNR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read '0' always. Note: Preselecting a Format Register. Writing to just the Control Field (using byte writes) does not initiate any SPI transfer in Master mode. This feature can be used to setup SPICLK Phase or Polarity before actually starting the transfer by just updating the DFSEL fields in the control field to select the required Phase/Polarity combination.
15-0	TXDATA	R/W	0h	SPI / MibSPI Transmit Data. When written, these bits will be copied to the Shift Register if it is empty. If the Shift Register is not empty, the TXBUF will hold the written values. SPIEN must be set to 1 before this register can be written to. Writing a 0 to the SPIEN register forces the lower 16 bits of the SPIDAT1 register to 0x00. Write to this register ONLY when using the automatic Slave Chip Select feature. See section 3, Operation Modes, on page 14. A write to this register will drive the contents of CSNR[7:0] into the respective pins in SPISCS[7:0] if those are configured as functional pins. When this register is read, contents of internal buffer register TXBUF which holds the latest written data will be returned. Irrespective of the character length, the Transmit data should be right justified before writing to SPIDAT1 register.

23.3.2.15 SPIBUF Register (Offset = 40h) [reset = 80000000h]

SPIBUF is shown in Figure 23-112 and described in Table 23-94.

Return to Summary Table.

SPI / MibSPI Receive Buffer Register

Figure 23-112. SPIBUF Register

31	30	29	28	27	26	25	24
RXEMPTY	RXOVR	TXFULL	BITERR	DESYNC	PARITYERR	TIMEOUT	DLENERR
1h	0h	R-0h	0h	0h	0h	0h	0h
23	22	21	20	19	18	17	16
LCSNR							
R-0h							
15	14	13	12	11	10	9	8
RXDATA							
R-0h							
7	6	5	4	3	2	1	0
RXDATA							
R-0h							

Table 23-94. SPIBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RXEMPTY		1h	Receive data buffer empty. When host reads the SPIBUF field or the whole SPIBUF register this will automatically set the RXEMPTY flag. When a data transfer is completed, the received data is copied into SPIBUF, the RXEMPTY flag is cleared. 1 =No data received since last reading of SPIBUF register. 0 =A new Data is received and copied into SPIBUF field. This flag gets set to '1' under following conditions: oReading RXDATA portion of the SPIBUF register. oWriting '1' to clear the RXINTFLG bit in SPIFLG register. Write-Clearing the RXINTFLG bit before reading the SPIBUF indicates the received data is being ignored. Conversely, RXINTFLG can be cleared by reading the RXDATA portion of the SPIBUF register. So, reading the full of SPIBUF register itself clears the Receiver Full Interrupt Flag.
30	RXOVR		0h	Receive data buffer overrun. When a data transfer is completed and the received data is copied into the RXBUF while it is already full, RXOVR is set. Refer to Figure 1 for a view of internal logic diagram. An Overrun always occurs to the RXBUF, and SPIBUF contents never get Overwritten until after it is read by the VBUSP Master. If enabled, RXOVRN interrupt gets generated when RXBUF is Overwritten, and reading SPIFLG or SPIVECTx register shows the RXOVRN condition. However, two read operations to the SPIBUF register are required to reach the Overrun buffer. 1 =A receive data overrun condition occurred since last time reading the data field. 0 =No receive data overrun condition occurred since last time reading the data field. This bit is cleared to '0' under the following conditions: o RXDATA portion of the SPIBUF register is read. o OVRNINTFLG bit in SPIFLG register is write-cleared. When an Overrun occurs, the SPIBUF contents will not be overwritten. Only the RXBUF contents are overwritten. When the SPIBUF is read out , the RXBUF contents get copied to the SPIBUF if RXBUF is full. So, the first data read out will be intact when an Overrun occurs. Note: A special condition under which RXOVR flag gets set. If both SPIBUF & RXBUF are already full and while another buffer receive is underway, if any errors like TIMEOUT, BITERR & DLEN_ERR occur, then RXOVR in RXBUF & SPIFLG registers will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a normal Receiver Overrun.

Table 23-94. SPIBUF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	TXFULL	R	0h	Transmit data buffer full. This flag is a read-only flag. Writing into SPIDAT0 or SPIDAT1 field while TX Shift Register is full will automatically set the TXFULL flag. Once the data is copied to the Shift Register the TXFULL flag will be cleared. Writing to SPIDAT0/SPIDAT1 register when both TXBUF & the TX Shift Register are empty does not set the TXFULL flag. 1 =Transmit buffer is full, SPIDAT0/SPIDAT1 is not ready to accept a new data. 0 =Transmit buffer is empty, SPIDAT0/SPIDAT1 is ready to accept a new data.
28	BITERR		0h	Mismatch of internal transmit data and transmitted data. 1 =A bit error occurred. The SPI / MibSPI samples the signal of the transmit pin (master: SIMO, slave: SOMI) at the receive point (half clock cycle after transmit point). If the sampled value differs from the transmitted value a bit error is detected and the flag BITERR is set. A possible reason for a bit error can be noise, a to high bit rate / capacitive load or another master/slave trying to transmit at the same time. 0 =No bit error occurred. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read.
27	DESYNC		0h	De-synchronization of slave device. De-synchronization monitor is active in master mode only. 1 =A slave device is de-synchronized. The master monitors the ENA signal coming from the slave device and sets the DESYNC flag if ENA is deactivated before the last reception point or after the last bit is transmitted plus tT2EDELAY (see Section 8.21). If DESYNCENA is set an interrupt is asserted. De-synchronization can occur if a slave device misses a clock edge coming from the master. 0 =No slave de-synchronization detected. This bit is valid in Master mode only. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read. Note: Possible inconsistency of Desync flag in SPI/Compatibility mode MibSPI Due to the nature of this Error, under some circumstances it is possible for Desync error detected for the previous buffer to be visible in the current buffer. This is due to the fact that Receive Completion flag/interrupt will be generated when the buffer transfer is completed. But Desync will be detected after the buffer transfer is completed. So, if VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct Desync condition. This inconsistency in Desync flag is valid only in SPI or Compatibility mode of MibSPI. In Multibuffer mode, Desync flag is always guaranteed to be for the current buffer.
26	PARITYERR		0h	Calculated parity differs from received parity bit. 1 =A parity error occurred. If the parity generator is enabled (can be selected individually for each buffer) an even or odd parity bit is added at the end of a data word (see Section 8.23). During reception of the data word the parity generator calculates the reference parity and compares it to the received parity bit. In the event of a mismatch the PARITYERR flag is set. 0 =No parity error detected. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read.
25	TIMEOUT		0h	Time-out due to non-activation of ENA pin. 1 =An ENA signal time-out occurred. The SPI / MibSPI generates a time-out because the slave hasn’t responded in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition the TIMOUT flag in the status field of the corresponding buffer and in the SPIFLG register is set. 0 =No ENA-pin time-out occurred. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read. This bit is valid in Master mode only.
24	DLENERR		0h	Data Length Error flag. 1 = A Data Length Error has occured. 0 = No Data Length Error has occured. This flag is cleared to ‘0’ when RXDATA portion of the SPIBUF register is read.

Table 23-94. SPIBUF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-16	LCSNR	R	0h	Last Chip select number. LCSNR in the status field is a copy of CSNR in the corresponding control field. It defines the chip select that has been activated during the last data transfer from the corresponding buffer. This is the copy of CSNR bits from SPIDAT1 latched at the end of a transfer. Note: Effect of NUM_CS_PINS generic on LCSNR bits. Actual number of bits implemented in LCSNR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will read ‘0’ always
15-0	RXDATA	R	0h	SPI Receive Data. This is the received data, transferred from the Receive Shift-Register at the end of a transfer completion. Irrespective of the programmed character length & the direction of shifting, the received data is stored right-justified in the register.

23.3.2.16 SPIEMU Register (Offset = 44h) [reset = 80000000h]

SPIEMU is shown in Figure 23-113 and described in Table 23-95.

Return to Summary Table.

SPI / MibSPI Emulation Register Note: All the fields ot SPIEMU register are Read-Only. Read operation on this register under any mode will not have any impact on the status of this or any other registers.

Figure 23-113. SPIEMU Register

31	30	29	28	27	26	25	24
RXEMPTY	RXOVR	TXFULL	BITERR	DESYNC	PARITYERR	TIMEOUT	DLENERR
R-1h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
LCSNR							
R-0h							
15	14	13	12	11	10	9	8
RXDATA							
R-0h							
7	6	5	4	3	2	1	0
RXDATA							
R-0h							

Table 23-95. SPIEMU Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RXEMPTY	R	1h	Receive data buffer empty. 1 = No data received since last reading of SPIBUF register. 0 = A new Data is received and copied into SPIBUF field.
30	RXOVR	R	0h	Receive data buffer overrun. 1 =A receive data overrun condition occurred since last time reading the data field. 0 =No receive data overrun condition occurred since last time reading the data field.
29	TXFULL	R	0h	Transmit data buffer full. 1 =Transmit buffer is full, SPIDAT0/SPIDAT1 is not ready to accept a new data. 0 =Transmit buffer is empty, SPIDAT0/SPIDAT1 is ready to accept a new data.
28	BITERR	R	0h	Mismatch of internal transmit data and transmitted data. 1 =A bit error occurred. The SPI / MibSPI samples the signal of the transmit pin (master: SIMO, slave: SOMI) at the receive point (half clock cycle after transmit point). If the sampled value differs from the transmitted value a bit error is detected and the flag BITERR is set. A possible reason for a bit error can be noise, a to high bit rate / capacitive load or another master/slave trying to transmit at the same time. 0 =No bit error occurred.
27	DESYNC	R	0h	De-synchronization of slave device. De-synchronization monitor is active in master mode only. 1 =A slave device is de-synchronized. The master monitors the ENA signal coming from the slave device and sets the DESYNC flag if ENA is deactivated before the last reception point or after the last bit is transmitted plus tT2EDELAY (see Section 8.21). If DESYNCENA is set an interrupt is asserted. De-synchronization can occur if a slave device misses a clock edge coming from the master. 0 =No slave de-synchronization detected. This bit is valid in Master mode only.
26	PARITYERR	R	0h	Calculated parity differs from received parity bit. 1 =A parity error occurred. If the parity generator is enabled (can be selected individually for each buffer) an even or odd parity bit is added at the end of a data word (see Section 8.23). During reception of the data word the parity generator calculates the reference parity and compares it to the received parity bit. In the event of a mismatch the PARITYERR flag is set. 0 =No parity error detected.

Table 23-95. SPIEMU Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	TIMEOUT	R	0h	Time-out due to non-activation of ENA pin. 1 =An ENA signal time-out occurred. The SPI / MibSPI generates a time-out because the slave hasn't responded in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition the TIMOUT flag in the status field of the corresponding buffer and in the SPIFLG register is set. 0 =No ENA-pin time-out occurred. This bit is valid in Master mode only.
24	DLENERR	R	0h	Data Length Error flag. 1 = A Data Length Error has occured. 0 = No Data Length Error has occured.
23-16	LCSNR	R	0h	Last Chip select number. LCSNR in the status field is a copy of CSNR in the corresponding control field. It defines the chip select that has been activated during the last data transfer from the corresponding buffer. This is the copy of CSNR bits from SPIDAT1 latched at the end of a transfer. Note: Effect of NUM_CS_PINS generic on LCSNR bits. Actual number of bits implemented in LCSNR[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will read '0' always.
15-0	RXDATA	R	0h	SPI Receive Data. SPI / MibSPI emulation is a mirror of the SPIBUF register. The only difference between SPIEMU and SPIBUF is that a read from SPIEMU does not clear any of the status flags.

23.3.2.17 SPIDELAY Register (Offset = 48h) [reset = 0h]

SPIDELAY is shown in Figure 23-114 and described in Table 23-96.

Return to Summary Table.

SPI / MibSPI Delay Register

Figure 23-114. SPIDELAY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2TDELAY								T2CDELAY								T2EDELAY								C2EDELAY							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 23-96. SPIDELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	C2TDELAY	R/W	0h	Chip-select-active-to-transmit-start-delay. C2TDELAY is used in master mode only. It defines a setup time for the slave device that delays the data transmission from the chip select active edge by a multiple of VBUSPCLK cycles. ChipSelect-active-to-transmission delays between 2 to 257 VBUSPCLK cycles can be achieved. If Phase = '1', the delay between SCS fall-edge to the first edge of SPICLK will have an additional 0.5 SPICLK Period delay. This delay is as per the SPI protocol.
23-16	T2CDELAY	R/W	0h	Transmit-end-to-chip-select-inactive-delay. T2CDELAY is used in master mode only. It defines a hold time for the slave device that delays the chip select deactivation by a multiple of VBUSPCLK cycles after the last bit is transferred. T2CDELAY can be configured between 2 and 256 VBUSPCLK cycles. If Phase = '0', then between the last edge of SPICLK and rise-edge of SCS, there will be an additional delay of 0.5 SPICLK period. This is as per the SPI protocol Both C2TDELAY and T2CDELAY counters do not have any dependancy on SPIENA pin value. Even if the SPIENA pin is asserted by the Slave, Master will continue to delay the start of SPICLK until the C2TDELAY counter overflows. Similarly, even if the SPIENA pin is deasserted by the Slave, Master will continue to hold the SPISCS pins active until the T2CDELAY counter overflows. This way, it is guaranteed that the setup/hold times of the SPISCS pins is determined by the Delay timers alone. To achieve better throughput, it should be ensured that these two timers are kept at the minimum possible values.
15-8	T2EDELAY	R/W	0h	Transmit-data-finished-to-ENA-pin-inactive-time-out. T2EDELAY is used in master mode only. It defines a time-out value as a multiple of SPI clock before the ENAble signal has to become inactive and after the CS becomes inactive. The SPI clock depends on which data format is selected. If the slave device is missing one or more clock edges, it is becoming de-synchronized. Although the master has finished the data transfer the slave is still waiting for the missed clock pulses and the ENA signal isn't disabled. The T2EDELAY defines a time-out value that triggers the DESYNC flag, if the ENA signal isn't deactivated in time. DESYNC flag is set to indicate that the Slave device did not deassert its SPIENA pin in time to acknowledge that it has received all the bits of the sent character. If T2CDELAY is programmed a non-zero value, then T2EDELAY will start only after the T2CDELAY completes. This should be taken into consideration to determine an optimum value of T2EDELAY. Note: Zero T2EDELAY If T2EDELAY is not programmed or programmed to '0', then the Master SPI/MibSPI does not wait for SPIENA pin to be deasserted. It ignores the state of the SPIENA pin after the transmission is completed.

Table 23-96. SPIDELAY Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	C2EDELAY	R/W	0h	Chip-select-active-to-ENA-signal-active-time-out C2EDELAY is utilized only in master mode and it applies only if the addressed slave generates an ENA signal as a hardware handshake response. C2EDELAY defines the maximum time between the SPI / MibSPI activating the chip select signal and the addressed slave responding by activating the ENA signal. C2EDELAY defines a time-out value as a multiple of SPI clocks. The SPI clock depends on whether data format 0 or data format 1 is selected. If the slave device is not responding with the ENA signal before the time-out value is reached, the TIMEOUT flag in SPIFLG register is set and an interrupt is asserted if enabled. If a time-out occurs the MibSPI clears the transmit request of the timed-out buffer, sets the TIMEOUT flag for the current buffer and continues with the transfer of the next buffer in the sequence that is enabled. If C2TDELAY is programmed a non-zero value, then C2EDELAY will start only after the C2TDELAY completes. This should be taken into consideration to determine an optimum value of C2EDELAY. Note: Zero C2EDELAY If C2EDELAY is not programmed or programmed to ‘0’, then the Master SPI/MibSPI waits until the SPIENA pin to asserted before it can start the transfer. If the SPIENA pin is not asserted due to a malfunctioning Slave, the Master will wait forever. This might cause hang-up situation. it is recommended to program to C2EDELAY to a suitable value whenever SPIENA pin is used as functional.

23.3.2.18 SPIDEF Register (Offset = 4Ch) [reset = 1h]

SPIDEF is shown in Figure 23-115 and described in Table 23-97.

Return to Summary Table.

SPI / MibSPI Default Chip select Register

Figure 23-115. SPIDEF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								CSDEF0							
R-0h																								R/W-1h							

Table 23-97. SPIDEF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
7-0	CSDEF0	R/W	1h	Chip select default pattern. Master mode behavior. The CSDEFx bits are output to the chip select pins when no transmission are currently performed. It allows the user to set a chip select pattern which deselects all the SPI slaves. 1 =If CSDEFx is set to “1” the corresponding chip select is set to “1” while SPI/MibSPI is IDLE. 0 =If CSDEFx is set to “0” the corresponding chip select is set to “0” while SPI/MibSPI is IDLE. Note: Effect of NUM_CS_PINS generic on CSDEF bits. Actual number of bits implemented in CSDEF[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be treated as reserved, read-only and will read ‘0’ always.

23.3.2.19 SPIFMT0 Register (Offset = 50h) [reset = 0h]

SPIFMT0 is shown in Figure 23-116 and described in Table 23-98.

Return to Summary Table.

SPI / MibSPI Data Format Register 0

Figure 23-116. SPIFMT0 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-98. SPIFMT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-98. SPIFMT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.2.20 SPIFMT1 Register (Offset = 54h) [reset = 0h]

SPIFMT1 is shown in Figure 23-117 and described in Table 23-99.

Return to Summary Table.

SPI / MibSPI Data Format Register 1

Figure 23-117. SPIFMT1 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-99. SPIFMT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-99. SPIFMT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.2.21 SPIFMT2 Register (Offset = 58h) [reset = 0h]

SPIFMT2 is shown in Figure 23-118 and described in Table 23-100.

Return to Summary Table.

SPI / MibSPI Data Format Register 2

Figure 23-118. SPIFMT2 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-100. SPIFMT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-100. SPIFMT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.2.22 SPIFMT3 Register (Offset = 5Ch) [reset = 0h]

SPIFMT3 is shown in Figure 23-119 and described in Table 23-101.

Return to Summary Table.

SPI / MibSPI Data Format Register 3

Figure 23-119. SPIFMT3 Register

31	30	29	28	27	26	25	24
WDELAY							
R/W-0h							
23	22	21	20	19	18	17	16
PARPOL	PARITYENA	WAITENA	SHIFTDIR	HDUPLEX_ENA	DISCSTIMERS	POLARITY	PHASE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
PRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
NU			CHARLEN				
R-0h			R/W-0h				

Table 23-101. SPIFMT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WDELAY	R/W	0h	Delay in between transmissions for data format x (x= 0,1,2,3). Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: WDELAY * PVBUSPCLK + 2 * PVBUSPCLK. PVBUSPCLK -> Period of VBUSPCLK.
23	PARPOL	R/W	0h	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 1 =An odd parity flag is added at the end of the transmit data stream. 0 =An even parity flag is added at the end of the transmit data stream.
22	PARITYENA	R/W	0h	Parity enable for data format x. 1= A parity is transmitted at the end of each transmit data stream. At the end of a transfer the parity generator compares the received parity bit with the locally calculated parity flag. If the parity bits do not match the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 0= No parity generation/ verification is performed for this data format. If an Uncorrectable Error Flag is set in a Slave mode MibSPI, then wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SOMI pin will be forced to transmit all ‘0’s. And parity bit will be transmitted as ‘1’ if even parity is selected and as ‘0’ if odd parity is selected(using the PARPOLx bit of this register). This behavior will be irrespective of an UPE on either TXRAM or RXRAM.
21	WAITENA	R/W	0h	Master waits for ENA signal from slave for data format x. WAITENA is considered in master mode only. In slave mode this bit has no meaning. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines for each buffer whether the addressed slave generates the ENA signal or does not. 1= Before the SPI / MibSPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the Master aborts the transfer and sets the TIMEOUT error flag. 0= The SPI / MibSPI does not wait for the ENA signal from the slaves and directly starts the transfer.

Table 23-101. SPIFMT3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	SHIFTDIR	R/W	0h	Shift direction for data format x. With bit SHIFTDIRx the shift direction for data format x (x=0,1,2,3) can be selected. 1 =Data format x shift direction: Least significant bit is shifted out first. 0 =Data format x shift direction: Most significant bit is shifted out first.
19	HDUPLEX_ENA	R/W	0h	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin. 0 = Normal Full Duplex transfer. 1 = If MASTER = '1', SIMO pin will act as an RX pin (No TX possible) If MASTER = '0', SIMO pin will act as a TX pin (No RX possible). For all normal operations, HDUPLEX_ENAx bits should always remain '0'. It is intended for the usage when the SIMO pin is used for both TX & RX operations at different times.
18	DISCSTIMERS	R/W	0h	Disable Chipselect Timers for this format register. The C2TDELAY & T2CDELAY timers are by default enabled for all the Data Format registers. Using this bit, these timers can be disabled for particular Data Format if not required. When a Master is handling multiple Slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the ChipSelect Delay timers for any Slaves. 0 = Both C2TDELAY & T2CDELAY counts are inserted for the ChipSelects. 1 = No C2TDELAY or T2CDELAY is inserted in the ChipSelect timings.
17	POLARITY	R/W	0h	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. POLARITYx can be modified in privilege mode only. 1 =If POLARITYx is set to "1" the SPI clock signal is high-inactive, i.e. before and after data transfer the clock signal is high. 0 =If POLARITYx is set to "0" the SPI clock signal is low-inactive, i.e. before and after data transfer the clock signal is low.
16	PHASE	R/W	0h	SPI Data format x clock delay. PHASEx defines the clock delay of data format x. PHASEx can be modified in privilege mode only. 1 =If PHASEx is set to "1" the SPI clock signal is delayed by a half SPI clock cycle versus the transmit / receive data stream. The first transmit bit has to output prior to the first clock edge. Master and slave receive the first bit with the first edge 0 =If PHASEx is set to "0" the SPI clock signal is not delayed versus the transmit / receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge Note: Restriction on SPICLK Phase/Polarity change in Slave Mode In Slave mode if Phase and/or Polarity of SPICLK has to be changed, the following sequence should be used. oClear the GCR1.SPIEN bit to '0'. oSet the required Phase/Polarity values in SPIFMTx registers. oSet the GCR1.SPIEN bit back to '1'. The setting of GCR1.SPIEN bit in Slave SPI/MibSPI to '1' should be done only after the Polarity of the incoming SPICLK signal changes (if POLARITYx bit was changed in the configuration).
15-8	PRESCALE	R/W	0h	SPI data format x prescaler. PRESCALEx can be modified in privilege mode only. PRESCALEx determines the bit transfer rate of data format x if the SPI is the network master. PRESCALEx is directly derived from VBUSPCLK. If the SPI / MibSPI is configured as slave, PRESCALEx DOES NOT NEED to be configured. The clock rate for data format x can be calculated as BRFormat = VBUSPCLCK/(PRESCALEx+1) When PRESCALEx is set to zero (0), the SPI clock rate defaults to VBUSPCLK/2. Any write to this field will update EPRESCALE_FMTx field of EPRESCALEy (y=1,2) registers with EPRESCALE_FMTx(11:8) bits rounded off to '000'. Any write to EPRESCALE_FMTx field of the EXTENDED_PRESCALEy (y=1,2) register will cause its lower 8bits to be reflected in this field as well.
7-5	NU	R	0h	Reserved. Reads return '0' and writes have no effect.
4-0	CHARLEN	R/W	0h	SPI data format x data word length. CHARLENx defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 0x10 (data word length = 16). Illegal values, such as 0x00 or 0x1F are not detected and their effect is indeterminate.

23.3.2.23 TGINTVECT0 Register (Offset = 60h) [reset = 0h]

TGINTVECT0 is shown in Figure 23-120 and described in Table 23-102.

Return to Summary Table.

SPI Interrupt Vector Register 0 / MibSPI Transfer Group Interrupt Vector Register 0

Figure 23-120. TGINTVECT0 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU				INTVECT0			SUSPEND0
R-0h				R-0h			R-0h

Table 23-102. TGINTVECT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
5-1	INTVECT0	R	0h	Interrupt vector for interrupt line INT0. INTVECT0 returns the vector of the pending interrupt at interrupt line INT0. If more than one interrupts are pending, INTVECT0 always references the highest priority interrupt source first. The vectors generated are different for SPI / Compatibility mode of MibSPI and Multibuffer mode of MibSPI. INTVECT0 field just reflects the status of SPIFLG register in a vectorized format. So, any updates to SPIFLG will automatically reflect in the Vector value in this register. SPI / Compatibility mode - MibSPI The interrupts available for SPI or Compatibility mode - MibSPI, in the descending order of their priorities are as given below. Transmission Error Interrupt Receive Buffer Overrun Interrupt Receive Buffer Full Interrupt Transmit Buffer Empty Interrupt Vectors for each of these interrupts will be reflected on the INTVECT0 bits, when they occur. Reading the vectors for the “Receive Buffer Overrun” & “Receive Buffer Full” interrupts will automatically clear the respective flags in the SPIFLG register. On reading the INTVECT0 bits, the vector of the next highest priority interrupt (if any) will be then reflected on the INTVECT0 bits. If two or more interrupts occur simultaneously, the vector for the higest priority interrupt will be reflected on the INTVECT0 bits. Reading the Vector register when “Transmitter Empty” is indicated does not clear the TXINTFLG in SPIFLG register. Writing a new data to SPIDATx register clears the “Transmitter Empty” interrupt. INTVECT0[4:0] Description 00000b no interrupt pending 10001b Error interrupt pending. Refer to Least Significant (LS) Byte of SPIFLG to determine more details about the type of error. 10011b Pending interrupt is “Receive Buffer Overrun Interrupt” 10010b Pending interrupt is “Receive Buffer Full Interrupt” 10100b Pending interrupt is “Transmit Buffer Empty Interrupt” All other bit combinations Reserved Note: Exception for clearing of RXINT If both SPIBUF and RXBUF (internal buffer) are full, then, reading TGINTVECT0 register (while it shows 10010) does not clear the RXINTFLG in SPIFLG register. In this case, only way to clear the Interrupt is to read the SPIBUF again until there’s no more unread RX data.

Table 23-102. TGINTVECT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SUSPEND0	R	0h	“Transfer suspended” or “transfer finished” interrupt.(MibSPI only) The SUSPEND0 flag is updated depending on the type of interrupt reflected by the VECTOR value field. 1 =The interrupt type is a “transfer suspended” interrupt. I.e. the transfer group referenced by INTVECT0 has asserted an interrupt, because the buffer to be transferred next is in “suspend to wait” mode. 0 =The interrupt type is a “transfer finished” interrupt. I.e. the buffer array referenced by INTVECT0 has asserted an interrupt, because all of the data from the whole transfer group has been transferred. Note: Special case for SUSPEND interrupt When there’s a “transfer suspended” interrupt (SUSPEND0 bit is set to ‘1’), reading the TGINTVECT0 register does not clear the “TG Suspended” interrupt. The SUSPEND condition should be resolved first before the interrupt can be cleared. The SUSPEND condition can be cleared by writing a new data to that TXRAM location and/or reading the data from that RXRAM location depending upon the SUSPEND criteria programmed in the “BUFMODE” field of that TXRAM location. The SUSPEND0 bit always returns value ‘0’ in SPI / Compatibility mode MibSPI. When there is an RXOVRN or any Error interrupt in Multibuffer mode, SUSPEND0 bit stays ‘0’. Refer to Section 6.2 on page 77 & Section 6.3 on page 77 for more details and notes on better handling of interrupts. Note: Reading Error Vector Reading an Error Vector in the TGINTVECT0 register will NOT clear the Error flags in the SPIFLG register. The Error Flags in SPIFLG need to be write-cleared after servicing them suitably. If “TG Completed” interrupt occurs for a TG and after a while “TG Suspended” flag too gets set for the same TG, then the TGINTVECT0 register will show “TG Completed” interrupt giving it higher priority than the “TG Suspended” interrrupt.

23.3.2.24 TGINTVECT1 Register (Offset = 64h) [reset = 0h]

TGINTVECT1 is shown in Figure 23-121 and described in Table 23-103.

Return to Summary Table.

SPI Interrupt Vector Register 1 / MibSPI Transfer Group Interrupt Vector Register 1

Figure 23-121. TGINTVECT1 Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU		INTVECT1					SUSPEND1
R-0h		R-0h					R-0h

Table 23-103. TGINTVECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
5-1	INTVECT1	R	0h	Interrupt vector for interrupt line INT1. INTVECT1 returns the vector of the pending interrupt at interrupt line INT0. If more than one interrupt is pending, INTVECT1 always references the highest priority interrupt source first. The vectors generated are different for SPI / Compatibility mode of MibSPI and Multibuffer mode of MibSPI. INTVECT1 field just reflects the status of SPIFLG register in a vectorized format. So, any updates to SPIFLG will automatically reflect in the Vector value in this register. SPI / Compatibility mode - MibSPI The interrupts available for SPI / Compatibility mode - MibSPI, in the descending order of their priorities are as given below. Transmission Error Interrupt Receive Buffer Overrun Interrupt Receive Buffer Full Interrupt Transmit Buffer Empty Interrupt Vectors for each of these interrupts will be reflected on the INTVECT1 bits, when they occur. Reading the vectors for the “Receive Buffer Overrun” & “Receive Buffer Full” interrupts will automatically clear the respective flags in the SPIFLG register. On reading the INTVECT1 bits, the vector of the next highest priority interrupt (if any) will then be reflected on the INTVECT1 bits. If two or more interrupts occur simultaneously, the vector for the higest priority interrupt will be reflected on the INTVECT1 bits. Reading the Vector register when “Transmitter Empty” is indicated does not clear the TXINTFLG in SPIFLG register. Writing a new data to SPIDATx register clears the “Transmitter Empty” interrupt. 00000b no interrupt pending 10001b Error interrupt pending. Refer to LS Byte of SPIFLG to determine more details about the type of error. 10011b Pending interrupt is “Receive Buffer Overrun Interrupt” 10010b Pending interrupt is “Receive Buiffer Full Interrupt” 10100b Pending interrupt is “Transmit Buffer Empty Interrupt” All other bit combinations Reserved Note: Exception for clearing of RXINT If both SPIBUF and RXBUF (internal buffer) are full, then, reading TGINTVECT1 register (while it shows 10010) does not clear the RXINTFLG in SPIFLG register. In this case, only way to clear the Interrupt is to read out the SPIBUF again until there’s no more unread RX data.

Table 23-103. TGINTVECT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SUSPEND1	R	0h	“Transfer suspended” or “transfer finished” interrupt.(MibSPI Only) The SUSPEND1 flag is updated depending on the type of interrupt reflected by the VECTOR value field. 1 =The interrupt type is a “transfer suspended” interrupt. I.e. the transfer group referenced by INTVECT1 has asserted an interrupt, because the buffer to be transferred next is in “suspend to wait” mode. 0 =The interrupt type is a “transfer finished” interrupt. I.e. the buffer array referenced by INTVECT1 has asserted an interrupt, because all data from the whole transfer group has been transferred. Note: Special case for SUSPEND interrupt When there’s a “transfer suspended” interrupt(SUSPEND1 bit is set to ‘1’), reading the TGINTVECT1 register does not clear the “TG Suspended” interrupt. The SUSPEND condition should be resolved first before the interrupt can be cleared. This condition can be cleared by writing a new data to that TXRAM location and/or reading the data from that RXRAM location depending upon the SUSPEND criteria programmed in the “BUFMODE” field of that TXRAM location. The SUSPEND1 bit always returns value ‘0’ in SPI / Compatibility mode MibSPI. Even while there is an RXOVRN or any Error interrupt in Multibuffer mode, SUSPEND1 bit stays ‘0’. Refer to Section 6.2 on page 77 & Section 6.3 on page 77 for more details and notes on better handling of interrupts. Note: Reading Error Vector Reading an Error Vector in the TGINTVECT1 register will NOT clear the Error flags in the SPIFLG register. The Error Flags in SPIFLG need to be write-cleared after servicing them suitably. If “TG Completed” interrupt occurs for a TG and after a while “TG Suspended” flag too gets set for the same TG, then the TGINTVECT1 register will show “TG Completed” interrupt giving it higher priority than the “TG Suspended” interrrupt.

23.3.2.25 SPIPC9 Register (Offset = 68h) [reset = 0h]

SPIPC9 is shown in Figure 23-122 and described in Table 23-104.

Return to Summary Table.

SPI/MibSPI Pin Control Register 9 (SPIPC9) - SPISRSEL

Figure 23-122. SPIPC9 Register

31	30	29	28	27	26	25	24
SOMISRS7							
R/W-0h							
23	22	21	20	19	18	17	16
SIMOSRS7							
R/W-0h							
15	14	13	12	11	10	9	8
NU				SOMISRS0	SIMOSRS0	CLKSRS	ENASRS
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
SCSSRS							
R/W-0h							

Table 23-104. SPIPC9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SOMISRS7	R/W	0h	Each of these 7 bits controls the slew rate for the corresponding SPISOMIx pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select.
23-16	SIMOSRS7	R/W	0h	Each of these 7 bits controls the slew rate for the corresponding SPISIMOx pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select.
15-12	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
11	SOMISRS0	R/W	0h	This bit controls the slew rate for SPISOMI0 pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select. Note: Bit 11 or bit 24 can be used to control the slew rate for SPISOMI0. If a 32 bit write is performed, bit 11 will have priority over bit 24.
10	SIMOSRS0	R/W	0h	This bit controls the slew rate for SPISIMO0 pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select. Note: Bit 10 or bit 16 can be used to control the slew rate for SPISIMO0. If a 32 bit write is performed, bit 10 will have priority over bit 16.
9	CLKSRS	R/W	0h	This bit controls the slew rate for SPICLK pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select.
8	ENASRS	R/W	0h	This bit controls the slew rate for SPIENA pin. 0 =Fast Buffer Select. 1 =Slow Buffer Select.
7-0	SCSSRS	R/W	0h	Each of these 7 bits controls the slew rate for the corresponding SPISCSx pin. 0 =Normal Buffer Select. 1 =Slow Buffer Select. Note: Effect of NUM_CS_PINS generic on ChipSelect bits. Actual number of bits implemented in SCSSRS[7:0] will depend upon the NUM_CS_PINS generic set during synthesis. Unimplemented bits will be read-only and will read ‘0’ always.

23.3.2.26 SPIPMCTRL Register (Offset = 6Ch) [reset = 0h]

SPIPMCTRL is shown in Figure 23-123 and described in Table 23-105.

Return to Summary Table.

SPI/MibSPI Parallel/Modulo Mode Control Register

Figure 23-123. SPIPMCTRL Register

31	30	29	28	27	26	25	24
NU4	HSM_MODE3	MODCLKPOL3	MMODE3			PMODE3	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	
23	22	21	20	19	18	17	16
NU3	HSM_MODE2	MODCLKPOL2	MMODE2			PMODE2	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
NU2	HSM_MODE1	MODCLKPOL1	MMODE1			PMODE1	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	
7	6	5	4	3	2	1	0
NU1	HSM_MODE0	MODCLKPOL0	MMODE0			PMODE0	
R-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

Table 23-105. SPIPMCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
30	HSM_MODE3	R/W	0h	High Speed Modulo Mode control bit for Data Format 3. Controls whether the PMODE3 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE3 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE3 bits are non-zero
29	MODCLKPOL3	R/W	0h	Modulo mode SPICLK Polarity for Data Format 3 Determines the Polarity of the SPICLK in Modulo mode only. If MODULO MODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected
28-26	MMODE3	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 3.000 = Normal single dataline mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
25-24	PMODE3	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 3. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”) 10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)
23	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
22	HSM_MODE2	R/W	0h	High Speed Modulo Mode control bit for Data Format 2. Controls whether the PMODE2 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE2 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE2 bits are non-zero

Table 23-105. SPIPMCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	MODCLKPOL2	R/W	0h	Modulo mode SPICLK Polarity for Data Format 2. Determines the Polarity of the SPICLK in Modulo mode only. If MMODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected
20-18	MMODE2	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 2. 000 = 1-data line Mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
17-16	PMODE2	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 2. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”)10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)
15	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
14	HSM_MODE1	R/W	0h	High Speed Modulo Mode control bit for Data Format 1. Controls whether the PMODE1 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE1 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE1 bits are non-zero.
13	MODCLKPOL1	R/W	0h	Modulo mode SPICLK Polarity for Data Format 1. Determines the Polarity of the SPICLK in Modulo mode only. If MMODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected.
12-10	MMODE1	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 1. 000 = 1-data line Mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
9-8	PMODE1	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 1. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”) 10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)
7	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
6	HSM_MODE0	R/W	0h	High Speed Modulo Mode control bit for Data Format 0. Controls whether the PMODE0 bits will result in Modulo Format data transfer or not. Refer to Section 3.26 for details about the HSM Mode. 0 = Normal mode - Normal Parallel mode if PMODE0 bits are non-zero. 1 = High Speed Modulo Mode. Data transfer will happen in Modulo Format if PMODE0 bits are non-zero.
5	MODCLKPOL0	R/W	0h	Modulo mode SPICLK Polarity for Data Format 0. Determines the Polarity of the SPICLK in Modulo mode only. If MMODE[2:0] bits are “000”, this bit will be ignored. 0 = Normal SPICLK in all the modes. 1 = Polarity of the SPICLK will be inverted if Modulo mode is selected.

Table 23-105. SPIPMCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-2	MMODE0	R/W	0h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if Modulo Option is supported by the module) for Data Format 0. 000 = 1-data line Mode - Default (PMODE should be set to “00”) 001 = 2-data line Mode (PMODE should be set to “00”) 010 = 3-data line mode (PMODE should be set to “00”) 011 = 4-data line mode (PMODE should be set to “00”) 100 = 5-data line mode (PMODE should be set to “00”) 101 = 6-data line mode (PMODE should be set to “01”) 110 = Reserved 111 = Reserved
1-0	PMODE0	R/W	0h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2 ,4 or 8 data lines for Data Format 0. 00 = normal operation / 1-data line (MMODE should be set to “000”) 01 = 2-data line mode (MMODE should be set to “000”) 10 = 4-data line mode (MMODE should be set to “000”) 11 = 8-data line mode (MMODE should be set to “000”)

23.3.2.27 MIBSPIE Register (Offset = 70h) [reset = 500h]

MIBSPIE is shown in Figure 23-124 and described in Table 23-106.

Return to Summary Table.

MibSPI Enable Register

Figure 23-124. MIBSPIE Register

31	30	29	28	27	26	25	24
NU3							
R-0h							
23	22	21	20	19	18	17	16
NU3							RXRAMACCESS
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU2				EXTENDED_BUF_ENA			
R-0h				R/W-5h			
7	6	5	4	3	2	1	0
NU1							MSPIENA
R-0h							R/W-0h

Table 23-106. MIBSPIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
16	RXRAMACCESS	R/W	0h	Receive RAM Access control Bit. During normal operating mode of MibSPI, the Receive Data/Status portion of Multibuffer RAM is read-only. To enable testing of Data Integrity checks of Receive RAM, a special read/write access control is provided through this bit. 0 = The RX portion of Multibuffer RAM is not writable by the CPU. That is, portion of Multibuffer RAM, addressed by an offset of 0x200-0x3FF is write protected. 1 = The whole of Multibuffer RAM is fully accessible for read/write by the CPU.
15-12	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect
11-8	EXTENDED_BUF_ENA	R/W	5h	Enables the support for 256 buffers. By default MibSPI supports up to 128 buffers for both TX and RX. It is also possible to extend the support to 256 buffers as a parameterized implementation. This field can be used to enable/disable the support for Extended Buffers. This Enable field is implemented only if “EXTENDED_BUF” parameter is set to ‘1’. If the parameter is set to ‘0’, this field is read-only and reads the disable value. Write (Privilege mode only) 1010 - Enable the Extended Buffer mode - up to 256 buffers can be used 0101 - Disable the Extended Buffer mode - MibSPI supports only 128 buffers All other values - writes are ignored and the values are not updated into this field. The state of the feature remains unchanged. Read (both privilege and user modes) 1010 - Extended Buffer mode is enabled - up to 256 buffers can be used 0101 - Extended Buffer mode is disabled - MibSPI supports only 128 buffers
7-1	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.

Table 23-106. MIBSPIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	MSPIENA	R/W	0h	Multibuffer mode Enable. After power-up or reset MSPIENA remains cleared, which means that the MibSPI runs in compatibility mode by default. If Multibuffer mode is desired, this register should be configured first after configuring the SPIGCR0 register. Unless MSPIENA is set to '1', the Multibuffer mode registers are not writable. Refer to Section 3.3 for the grouping of registers into Compatibility mode and Multibuffer mode. 1 =The MibSPI is configured to run in MibSPI mode (Multibuffer mode). In this mode the additional features are available. 0 =The MibSPI runs in compatibility mode, i.e. in this mode the MibSPI is fully code compliant to the standard TMS470 Platform SPI. No Multibuffer feature is supported

23.3.2.28 TGITENST Register (Offset = 74h) [reset = 0h]

TGITENST is shown in Figure 23-125 and described in Table 23-107.

Return to Summary Table.

MibSPI Transfer Group Interrupt Enable Set Register

Figure 23-125. TGITENST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETINTENRDY																SETINTENSUS															
R/W-0h																R/W-0h															

Table 23-107. TGITENST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SETINTENRDY	R/W	0h	Transfer group interrupt set (enable) when transfer finished. Write: 1 = Enables the “The Transfer group x completed “ interrupt Interrupt gets generated when Transfer Group x gets completed. 0 = Has no effect. Read: 1 = “The Transfer group x completed “ interrupt is enabled Interrupt gets generated when Transfer Group x gets completed. 0 = “The Transfer group x completed” interrupt is disabled Interrupt does not get generated when Transfer Group x gets completed
15-0	SETINTENSUS	R/W	0h	Transfer group interrupt set (enable) when transfer suspended Write: 1 = Enables the “The Transfer group x suspended “ interrupt Interrupt gets generated when Transfer Group x gets suspended. 0 = Has no effect. Read: 1 = “The Transfer group x suspended “ interrupt is enabled Interrupt gets generated when Transfer Group x gets suspended. 0 = “The Transfer group x suspended” interrupt is disabled Interrupt does not get generated when Transfer Group x gets suspended

23.3.2.29 TGITENCR Register (Offset = 78h) [reset = 0h]

TGITENCR is shown in Figure 23-126 and described in Table 23-108.

Return to Summary Table.

MibSPI Transfer Group Interrupt Enable Clear Register

Figure 23-126. TGITENCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRINTENRDY																CLRINTENSUS															
R/W-0h																R/W-0h															

Table 23-108. TGITENCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CLRINTENRDY	R/W	0h	Transfer group interrupt clear (disable) when transfer finished. Write: 1 = Disables the “The Transfer group x completed “ interrupt Interrupt does not get generated when Transfer Group x gets completed. 0 = Has no effect. Read: 1 = “The Transfer group x completed “ interrupt is enabled Interrupt gets generated when Transfer Group x gets completed. 0 = “The Transfer group x completed” interrupt is disabled Interrupt does not get generated when Transfer Group x gets completed
15-0	CLRINTENSUS	R/W	0h	Transfer group interrupt clear (disable) when transfer suspended Write: 1 = Disables the “The Transfer group x suspended “ interrupt Interrupt does not get generated when Transfer Group x gets suspended. 0 = Has no effect. Read: 1 = “The Transfer group x suspended “ interrupt is enabled Interrupt gets generated when Transfer Group x gets suspended. 0 = “The Transfer group x suspended” interrupt is disabled Interrupt does not get generated when Transfer Group x gets suspended

23.3.2.30 TGITLVST Register (Offset = 7Ch) [reset = 0h]

TGITLVST is shown in Figure 23-127 and described in Table 23-109.

Return to Summary Table.

MibSPI Transfer Group Interrupt Level Set Register

Figure 23-127. TGITLVST Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETINTLVLRDY																SETINTLVLSUS															
R/W-0h																R/W-0h															

Table 23-109. TGITLVST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SETINTLVLRDY	R/W	0h	Transfer group completed" Interrupt Level set register Write: 1 = Sets the "The Transfer group x completed " interrupt to line INT1 0 = Has no effect. Read: 1 = "The Transfer group x completed " interrupt is set to line INT1 0 = "The Transfer group x completed " interrupt is set to line INT0
15-0	SETINTLVLSUS	R/W	0h	Transfer group suspended" interrupt Level set rigester Write: 1 = Sets the "The Transfer group x suspended " interrupt to line INT1 0 = Has no effect. Read: 1 = "The Transfer group x suspended " interrupt is set to line INT1 0 = "The Transfer group x suspended " interrupt is set to line INT0

23.3.2.31 TGITLVCR Register (Offset = 80h) [reset = 0h]

TGITLVCR is shown in Figure 23-128 and described in Table 23-110.

Return to Summary Table.

MibSPI Transfer Group Interrupt Level Clear Register

Figure 23-128. TGITLVCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRINTLVLRDY																CLRINTLVLSUS															
R/W-0h																R/W-0h															

Table 23-110. TGITLVCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CLRINTLVLRDY	R/W	0h	Transfer group completed" Interrupt Level clear register Write: 1 = Sets the "The Transfer group x completed " interrupt to line INT0 0 = Has no effect. Read: 1 = "The Transfer group x completed " interrupt is set to line INT1 0 = "The Transfer group x completed " interrupt is set to line INT0
15-0	CLRINTLVLSUS	R/W	0h	Transfer group suspended" interrupt Level clear register Write: 1 = Sets the "The Transfer group x suspended " interrupt to line INT0 0 = Has no effect. Read: 1 = "The Transfer group x suspended " interrupt is set to line INT1 0 = "The Transfer group x suspended " interrupt is set to line INT0

23.3.2.32 TGINTFLAG Register (Offset = 84h) [reset = 0h]

TGINTFLAG is shown in Figure 23-129 and described in Table 23-111.

Return to Summary Table.

Transfer Group Interrupt Flag Register

Figure 23-129. TGINTFLAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTFLGRDY																INTFLGSUS															
R-0h																R-0h															

Table 23-111. TGINTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	INTFLGRDY	R	0h	Transfer group interrupt flag for “transfer finished” interrupt. Read: 1 =A “transfer finished” interrupt from transfer group x occurred. No matter whether the interrupt is enabled or disabled (INTENRDYx = don’t care) or whether the interrupt is mapped to line INT0 or INT1, INTFLGRDYx is set right after the transfer from transfer group x is finished. 0 =No “transfer finished” interrupt occurred since last clearing of the flag INTFLGRDYx. Write: 1 = Clears the corresponding bit flag. 0 = Has no effect.
15-0	INTFLGSUS	R	0h	Transfer group interrupt flag for “transfer suspend” interrupt. Read: 1 =A “transfer suspended” interrupt from transfer group x occurred. No matter whether the interrupt is enabled or disabled (INTENSUSx = don’t care) or whether the interrupt is mapped to line INT0 or INT1, INTFLGSUSx is set right after the transfer from transfer group x is suspended. 0 =No ”transfer suspended” interrupt occurred since last clearing of the flag INTFLGSUSx. Note: Read Clear Behavior Reading the interrupt vector registers TGINTVECT0 or TGINTVECT1 automatically clears the interrupt flag bit INTFLGRDYx referenced by the vector number given by INTVECT0/INTVECT1 bits, if SUSPEND0/SUPEND1 bit in the Vector registers is ‘0’.

23.3.2.33 TICKCNT Register (Offset = 90h) [reset = 0h]

TICKCNT is shown in Figure 23-130 and described in Table 23-112.

Return to Summary Table.

Tick Count Register

Figure 23-130. TICKCNT Register

31	30	29	28	27	26	25	24
TICKENA	RELOAD	CLKCTRL		NU			
R/W-0h	R-0/W-0h	R/W-0h		R-0h			
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
TICKVALUE							
R/W-0h							
7	6	5	4	3	2	1	0
TICKVALUE							
R/W-0h							

Table 23-112. TICKCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TICKENA	R/W	0h	Tick counter enable. 1 =The MibSPI internal tick counter is enabled and is clocked by the clock source selected by CLKCTRL[1:0]. When the tick counter is enabled it starts down-counting from its current value. When TICKENA goes from “0” to “1” the tick counter is automatically loaded with the TICKVALUE. 0 =The MibSPI internal tick counter is disabled. The counter value remains unchanged. Note: When the tick counter is disabled the trigger signal is forced low.
30	RELOAD	R-0/W	0h	Re-load tick counter. RELOAD is a set-only bit, i.e. writing a “1” to it automatically reloads the Tick Counter with the value stored in TICKVALUE. Reading RELOAD always returns a “0”. Note: When the tick counter is reloaded by the RELOAD bit, the trigger signal is not toggled.
29-28	CLKCTRL	R/W	0h	Tick counter clock source control. CLKCTRL[1:0] defines the clock source that is used to clock the MibSPI internal tick counter.CLKCTRL[1:0] Description 00b SPICLK of Data word format 0 is selected as clock source of tick counter 01b SPICLK of Data word format 1 is selected as clock source of tick counter 10b SPICLK of Data word format 2 is selected as clock source of tick counter 11b SPICLK of Data word format 3 is selected as clock source of tick counter
27-16	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
15-0	TICKVALUE	R/W	0h	Initial value for tick counter. TICKVALUE stores the initial value for the tick counter. The tick counter is loaded with TICKVALUE every time an under-flow condition occurs and every time the RELOAD flag is set by the host

23.3.2.34 LTGPEND Register (Offset = 94h) [reset = 0h]

LTGPEND is shown in Figure 23-131 and described in Table 23-113.

Return to Summary Table.

Last Transfer Group End Pointer

Figure 23-131. LTGPEND Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU3			TGINSERVICE					NU2							
R-0h			R-0h					R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPEND								NU1							
R/W-0h								R-0h							

Table 23-113. LTGPEND Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU3	R	0h	Reserved.Reads return ‘0’ and writes have no effect
28-24	TGINSERVICE	R	0h	Transfer Group currently being serviced by the Sequencer. Read-Only field indicating the current Transfer Group that is being serviced. This field can generally be used for code debug purpose. Read Value:TG IN SERVICE[4:0] Description 00000b No Transfer Group is being serviced by the Sequencer 00001b Transfer Group0 is being serviced by the Sequencer ... ... 10000b Transfer Group15 is being serviced by the Sequencer 10001b - 11111b Invalid values
23-16	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect
15-8	LPEND	R/W	0h	Last Transfer Group End Pointer Usually the transfer group end address (PEND) is inherently defined by the start value of the starting pointer of the subsequent transfer group (PSTART). The transfer group ends at the buffer one before the next transfer group starts (PEND[x]=PSTART[x+1] - 1). For a full configuration of MibSPI, the 15th transfer group has no subsequent transfer group, i.e. no end address is inherently defined. Therefore LPEND has to be programmed to specify explicitly the end address of the 15th transfer group. Number of Transfer Groups implemented in a MibSPI can vary from one MibSPI to another since it is a generic parameter based implementation. If in a MibSPI, only 4 Transfer Groups are implemented, then the PEND of the 4th TG is defined by LPEND values defined in this LTGPEND register.
7-0	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect

23.3.2.35 TG0CTRL Register (Offset = 98h) [reset = 0h]

TG0CTRL is shown in Figure 23-132 and described in Table 23-114.

Return to Summary Table.

MibSPI Transfer Group Control Register The number of transfer groups is scalable by design up to a maximum of 16. Depending on the implementation the number of transfer groups and hence the number of transfer group control register may vary. Each transfer group can be configured via one dedicated control register. The register description below shows one exemplary control register(x) which is identical for all transfer groups. E.g. the control register for transfer group 2 is named “TG2CTRL” and is located at address base0+98h+4*2.

Figure 23-132. TG0CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-114. TG0CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.

Table 23-114. TG0CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved

Table 23-114. TG0CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.36 TG1CTRL Register (Offset = 9Ch) [reset = 0h]

TG1CTRL is shown in Figure 23-133 and described in Table 23-115.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-133. TG1CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-115. TG1CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-115. TG1CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-115. TG1CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.37 TG2CTRL Register (Offset = A0h) [reset = 0h]

TG2CTRL is shown in Figure 23-134 and described in Table 23-116.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-134. TG2CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-116. TG2CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-116. TG2CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-116. TG2CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.38 TG3CTRL Register (Offset = A4h) [reset = 0h]

TG3CTRL is shown in Figure 23-135 and described in Table 23-117.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-135. TG3CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-117. TG3CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-117. TG3CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-117. TG3CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.39 TG4CTRL Register (Offset = A8h) [reset = 0h]

TG4CTRL is shown in Figure 23-136 and described in Table 23-118.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-136. TG4CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-118. TG4CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-118. TG4CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-118. TG4CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.40 TG5CTRL Register (Offset = ACh) [reset = 0h]

TG5CTRL is shown in Figure 23-137 and described in Table 23-119.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-137. TG5CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-119. TG5CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-119. TG5CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-119. TG5CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.41 TG6CTRL Register (Offset = B0h) [reset = 0h]

TG6CTRL is shown in Figure 23-138 and described in Table 23-120.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-138. TG6CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-120. TG6CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-120. TG6CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-120. TG6CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.42 TG7CTRL Register (Offset = B4h) [reset = 0h]

TG7CTRL is shown in Figure 23-139 and described in Table 23-121.

Return to Summary Table.

MibSPI Transfer Group Control Register

Figure 23-139. TG7CTRL Register

31	30	29	28	27	26	25	24
TGENA	ONESHOT	PRST	TGTD	NU			
R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h			
23	22	21	20	19	18	17	16
TRIGEVT				TRIGSRC			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
PSTART							
R/W-0h							
7	6	5	4	3	2	1	0
PCURRENT							
R-0h							

Table 23-121. TG7CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TGENA	R/W	0h	Transfer Group Enable. 1 =The corresponding transfer group is enabled. If the correct event (TRIGEVTx) occurs at the selected source (TRIGSRCx) a group transfer is initiated if no higher priority transfer group is in active transfer mode or if one or more higher priority Transfer Groups are in transfer suspend mode. If higher priority Transfer Groups (TG) are in transfer mode, then the newly enabled TG will wait till all of the higher priority TG transfers are completed. 0 =The corresponding transfer group is disabled. Disabling a transfer group while a transfer is ongoing, will finish the ongoing buffer transfer but not the whole group transfer
30	ONESHOT	R/W	0h	Single transfer for this Transfer Group group. 1 =A transfer from the corresponding transfer group will be performed only once (= one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENAx control bit will be cleared by the MibSPI and therefore no additional transfer can be triggered before the host enables the transfer group again. This oneshot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data. 0 =The corresponding transfer group initiates a transfer every time a trigger event occurs and TGENA is set.
29	PRST	R/W	0h	transfer group Pointer Reset mode. With PRST, the way of resolving trigger events during an ongoing transfer from the concerned transfer group can be configured. This bit is meaningful only for Level Triggered Transfer Groups. Edge triggered TGs cannot be re-started before their completion by another edge. PRST bit will have no effect on this behavior. 1 =The corresponding transfer group pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same transfer group is ongoing. I.e. every trigger event resets PCURRENTx no matter whether the concerned transfer group is in transfer mode or not. The trigger events have priority over the ongoing transfer. 0 =If a trigger event occurs during a transfer from the concerned transfer group, the event is ignored and is not stored internally. The transfer group transfer has priority over additional trigger events.

Table 23-121. TG7CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	TGTD	R	0h	Transfer group triggered. This bit is read-only. 1 =The transfer group has been triggered and is either currently in service or waiting for servicing. 0 =The corresponding transfer group has not been triggered or is no more waiting for service. Use the “TG IN SERVICE” field in LTGPEND register to determine the exact Transfer Group being currently serviced
27-24	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect
23-20	TRIGEVT	R/W	0h	Type of trigger event. After reset, the trigger event types of all transfer groups are set to inactive TypesTRIGEVTx[3:0] Type Description 0000b never 0001b rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0010b falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0011b both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer from the corresponding transfer group 0100b reserved 0101b high-active Repetitive group transfer while trigger is high: While the selected trigger source (TRIGSRCx) is at a logic high level (1) the group transfer is continued and at the end of one group transfer restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. 0110b low-active Repetitive group transfer while trigger is low: While the selected trigger source (TRIGSRCx) is at a logic low level (0) the group transfer is continued and at the end of one restarted at the beginning. If ONESHOTx is set the transfer is performed only once. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. 0111b always A repetitive group transfer will be performed. Setting ONESHOTx allows a software controlled single transfer modea a. By setting the TRIGSRC to 0000b, the TRIGEVT to ALWAYS (0111b) the ONESHOT bit to 1. This allows a software control trigger on this Transfer Group. Then by setting the TGENA bit, the Transfer Group is immediately triggered. . If ONESHOTx is cleared a continuous mode for this Transfer Group is selected.1xxxb reserved
19-16	TRIGSRC	R/W	0h	Trigger source. After reset the trigger sources of all transfer groups are disabled. Table 22. Trigger SourcesTRIGSRCx[3:0] Type Description 0000b disabled 0001b EXT0 MibSPI external trigger source 0. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0010b EXT1 MibSPI external trigger source 1. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0011b EXT2 MibSPI external trigger source 2. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 0100b EXT3 MibSPI external trigger source 3. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). ... ... 1110b EXT13 MibSPI external trigger source 13. Source has to be defined individually for each Microcontroller derivative (e.g. HET I/O channel, event pin, etc.). 1111b TICK MibSPI internal periodic event trigger. The tick counter can initiate periodic group transfers.
15-8	PSTART	R/W	0h	transfer group start address. PSTARTx stores the start address of the corresponding transfer group. The corresponding end address is inherently defined by the subsequent transfer groups start address minus one (PENDx[TGx] = PSTARTx[TGx+1]-1). PSTARTx is copied into PCURRENTx when: o the transfer group is enabled o the end of the transfer group is reached during a transfer o a trigger event occurs while PRST is set to 1

Table 23-121. TG7CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	PCURRENT	R	0h	transfer group pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127/255) of the buffer that is currently transferred or that will be transferred after a trigger event occurs (if PRSTx=0) or after the transfer group resumes from suspend to wait mode. If the transfer group switches mode from active transfer mode to “suspend to wait”, PCURRENTx keeps the address of the currently Suspended buffer. After the transfer group resumes from “suspend to wait” mode the next buffer will be transferred. I.e. no buffer data is multiply transferred or not at all transferred due to “suspend to wait” mode.

23.3.2.43 DMA0CTRL Register (Offset = D8h) [reset = 0h]

DMA0CTRL is shown in Figure 23-140 and described in Table 23-122.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-140. DMA0CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-122. DMA0CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-122. DMA0CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.2.44 DMA1CTRL Register (Offset = DCh) [reset = 0h]

DMA1CTRL is shown in Figure 23-141 and described in Table 23-123.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-141. DMA1CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-123. DMA1CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-123. DMA1CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.2.45 DMA2CTRL Register (Offset = E0h) [reset = 0h]

DMA2CTRL is shown in Figure 23-142 and described in Table 23-124.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-142. DMA2CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-124. DMA2CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-124. DMA2CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.2.46 DMA3CTRL Register (Offset = E4h) [reset = 0h]

DMA3CTRL is shown in Figure 23-143 and described in Table 23-125.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-143. DMA3CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-125. DMA3CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-125. DMA3CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.2.47 DMA4CTRL Register (Offset = E8h) [reset = 0h]

DMA4CTRL is shown in Figure 23-144 and described in Table 23-126.

Return to Summary Table.

MibSPI DMA Channel Control Register

Figure 23-144. DMA4CTRL Register

31	30	29	28	27	26	25	24
ONESHOT	BUFID						
R/W-0h	R/W-0h						
23	22	21	20	19	18	17	16
RXDMA_MAP				TXDMA_MAP			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RXDMAENA	TXDMAENA	NOBRK	ICOUNT				
R/W-0h	R/W-0h	R/W-0h	R/W-0h				
7	6	5	4	3	2	1	0
BUFID7	COUNTBIT17	COUNT					
R/W-0h	R-0h	R-0h					

Table 23-126. DMA4CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ONESHOT	R/W	0h	Auto-disable of DMA channel after ICOUNT+1 transfers. 1 =ONESHOTx allows a block transfer of defined length (ICOUNTx+1) mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx +1 transfers the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer. 0 =The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.
30-24	BUFID	R/W	0h	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the “suspend to wait until...” modes must be used (for more details refer to Section 8.53.1).
23-20	RXDMA_MAP	R/W	0h	Receive data DMA Request Map Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set to ‘1’, then RXDMA_MAPx[3:0] shall differ from TXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.
19-16	TXDMA_MAP	R/W	0h	Transmit data DMA channel Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. ne request line for receive data and the other for request line for transmit data. TXDMA_MAPx[3:0] defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel. If RXDMAENAx and TXDMAENAx are both set then TXDMA_MAPx[3:0] shall differ from RXDMA_MAPx[3:0] and shall differ from any other used physical DMA Request line. Otherwise unexpected interference may occur.

Table 23-126. DMA4CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RXDMAENA	R/W	0h	Receive data DMA channel enable. 1 =The physical DMA Request line for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The concerned buffer should be configured in the mode “skip until RXEMPTY is set” or “suspend to wait until RXEMPTY is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new receive data.
14	TXDMAENA	R/W	0h	Transmit data DMA channel enable. 1 =The physical DMA Request line for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The concerned buffer should be configured in the mode “skip until TXFULL is set” or “suspend to wait until TXFULL is set” in order to ensure synchronization between DMA controller and MibSPI sequencer. 0 =No DMA request upon new transmit data
13	NOBRK	R/W	0h	Non-interleaved DMA block transfer(Master mode only). 1 =NOBRKx ensures that ICOUNTx+1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx+1 transfers have been processed. E.g: this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD=1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI. Triggering of higher priority transfer groups or enabling of higher priority DMA channels will not interrupt NOBRK block transfer. 0 =The DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or transfer groups. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer
12-8	ICOUNT	R/W	0h	Initial Count of DMA transfers ICOUNTx[4:0] is used to preset the transfer counter COUNTx[4:0]. Every time COUNTx[4:0] hits zero it is reloaded with ICOUNTx[4:0]. The real number of transfer equals ICOUNTx[4:0] plus one. If ONESHOTx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx[4:0] defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
7	BUFID7	R/W	0h	Extended bit of BUFIDx field when Extended Buffer feature is implemented. This bit represents the 8th bit of BUFID field such that any buffers between 127-255 can be configured as DMA capable buffers
6	COUNTBIT17	R	0h	The 17th bit of COUNT field of DMAxCOUNT register. This bit is useful only when ICOUNTx in DMAxCOUNT register is programmed to be 0xFFFF. During all other values, this bit remains to be ‘0’.
5-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx[5:0] is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set.

23.3.2.48 ICOUNT0 Register (Offset = F8h) [reset = 0h]

ICOUNT0 is shown in Figure 23-145 and described in Table 23-127.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-145. ICOUNT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-127. ICOUNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.2.49 ICOUNT1 Register (Offset = FCh) [reset = 0h]

ICOUNT1 is shown in Figure 23-146 and described in Table 23-128.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-146. ICOUNT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-128. ICOUNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.2.50 ICOUNT2 Register (Offset = 100h) [reset = 0h]

ICOUNT2 is shown in Figure 23-147 and described in Table 23-129.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-147. ICOUNT2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-129. ICOUNT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.2.51 ICOUNT3 Register (Offset = 104h) [reset = 0h]

ICOUNT3 is shown in Figure 23-148 and described in Table 23-130.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-148. ICOUNT3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-130. ICOUNT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.2.52 ICOUNT4 Register (Offset = 108h) [reset = 0h]

ICOUNT4 is shown in Figure 23-149 and described in Table 23-131.

Return to Summary Table.

MibSPI DMAxCOUNT

Figure 23-149. ICOUNT4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICOUNT																COUNT															
R/W-0h																R-0h															

Table 23-131. ICOUNT4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ICOUNT	R/W	0h	Initial Number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits zero it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus one. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer
15-0	COUNT	R	0h	Actual number of remaining DMA transfer COUNTx is a read-only bit field. It comprises the actual number of DMA transfers that remain, until the DMA channel is disabled if ONESHOTx is set. Since the real COUNTx is always ICLOUNTx +1, the 17th bit of COUNTx is available on DMAxCTRL(6) bit.

23.3.2.53 DMACNTLEN Register (Offset = 118h) [reset = 0h]

DMACNTLEN is shown in Figure 23-150 and described in Table 23-132.

Return to Summary Table.

DMA LARGE COUNT register

Figure 23-150. DMACNTLEN Register

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							LARGE_COUNT
R-0h							R/W-0h

Table 23-132. DMACNTLEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
0	LARGE_COUNT	R/W	0h	0: Writes to the DMAxCTRL register will modify the ICOUNT value. Reading ICOUNT and COUNT can be done from the DMAxCTRL register. The DMAxCOUNT register should not be used since any write to this register will be overwritten by a subsequent write to DMAxCTRL register to set the TXDMAENA or RXDMAENA bits. 1: Writes to the DMAxCTRL register will not modify the ICOUNT value. The ICOUNT value must be written to in the DMAxCOUNT register before the RXDMAENA or TXDMAENA bits are set in the DMAxCTRL register. The DMAxCOUNT register should be used for reading COUNT or ICOUNT.

23.3.2.54 PAR_ECC_CTRL Register (Offset = 120h) [reset = 050A0005h]

PAR_ECC_CTRL is shown in Figure 23-151 and described in Table 23-133.

Return to Summary Table.

Parity/ECC Control Register

Figure 23-151. PAR_ECC_CTRL Register

31	30	29	28	27	26	25	24
NU4				SBE_EVT_EN			
R-0h				R/W-5h			
23	22	21	20	19	18	17	16
NU3				EDAC_MODE			
R-0h				R/W-Ah			
15	14	13	12	11	10	9	8
NU2							PTESTEN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
NU1				EDEN			
R-0h				R/W-5h			

Table 23-133. PAR_ECC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	NU4	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
27-24	SBE_EVT_EN	R/W	5h	Single Bit Error Event Enable This bit controls the generation of Error signaling (on MIBSPI_SBERR port) whenever a Single Bit Errors (SBE) is detected on TXRAM/RXRAM. This signal can be used to generate interrupt if required. Write: 0101 - Disable Error Event indication upon detection of SBE on TXRAM/RXRAM 1010 - Enable Error Event upon detection of SBE on TXRAM/RXRAM All other values - writes are ignored and the values are not updated into this field. The state of the feature remains unchanged. Read: Returns the current value of the field
23-20	NU3	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
19-16	EDAC_MODE	R/W	Ah	Error Detection And Correction Mode These bits determine whether Single Bit Errors (SBE) detected by the SECDED block will be corrected or not. Write: 0101 - Disable correction of SBE detected by the SECDED block 1010 - Enable correction of SBE detected by the SECDED block All other values - writes are ignored and the values are not updated into this field. The state of the feature remains unchanged. Read: Returns the current value of the field
15-9	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
8	PTESTEN	R/W	0h	Parity/ECC memory Test Enable. This bit, maps the parity/ecc bits corresponding to Multibuffer RAM locations into the peripheral RAM frame to make them accessible by the CPU. User and privilege mode (read): 0 = parity/ecc bits are not memory mapped 1 = parity/ecc bits are memory mapped Privilege mode (write): 0 = disable memory mapping of Parity/ECC locations 1 = enable memory mapping of Parity/ECC locations
7-4	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
3-0	EDEN	R/W	5h	Error Detection Enable These bits enable Parity/ECC Error Detection. Write: 0101: Disables Parity/ECC Error Detection Logic(default) Others : Enables Parity/ECC Error Detection Logic. Read: Returns the current value of this field

23.3.2.55 PAR_ECC_STAT Register (Offset = 124h) [reset = 0h]

PAR_ECC_STAT is shown in Figure 23-152 and described in Table 23-134.

Return to Summary Table.

Parity/ECC Status Register

Figure 23-152. PAR_ECC_STAT Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2							
R-0h							
15	14	13	12	11	10	9	8
NU2						SBE_FLG1	SBE_FLG0
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
NU1						UERR_FLG1	UERR_FLG0
R-0h						R-0h	R-0h

Table 23-134. PAR_ECC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
9	SBE_FLG1	R	0h	Single Bit Error in RXRAM. This flag indicates if a single bit ECC Error ocurred on reading RXRAM Read: 0 = No error occured. 1 = Single bit error is detected in RXRAM and the address is captured in SBERRADDR1 register. Write: 0 = No effect. 1 = Clears the bit.
8	SBE_FLG0	R	0h	Single Bit Error in TXRAM. This flag indicates if a single bit ECC Error ocurred on reading TXRAM Read: 0 = No error occured. 1 = Single bit error is detected in TXRAM and the address is captured in SBERRADDR0 register. Write: 0 = No effect. 1 = Clears the bit.
7-2	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
1	UERR_FLG1	R	0h	Uncorrectable Parity or double bit ECC error detection flag This flag indicates if a Parity or double bit ECC error ocurred on reading RXRAM When this bit is read: 0 = No error occured. 1 = Error detected and the address is captured in UERRADDR1 register. When write to this bit with: 0 = No effect. 1 = Clears the bit
0	UERR_FLG0	R	0h	Uncorrectable Parity or double bit ECC error detection flag This flag indicates if a Parity or ECC error ocurred on reading TXRAM When this bit is read: 0 = No error occured. 1 = Error detected and the address is captured in UERRADDR0 register. When write to this bit with: 0 = No effect. 1 = Clears the bit.

23.3.2.56 UERRADDR1 Register (Offset = 128h) [reset = 0h]

UERRADDR1 is shown in Figure 23-153 and described in Table 23-135.

Return to Summary Table.

Uncorrectable Parity or double bit ECC error Address Register - RXRAM

Figure 23-153. UERRADDR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																					UERRADDR1										
R-0h																					R-0h										

Table 23-135. UERRADDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	UERRADDR1	R	0h	Uncorrectable Parity or double bit ECC error address This register holds the address of the RAM location if a parity or double bit ECC error is detected when reading the MibSPI (Receive) RXRAM. The address captured is byte alligned when RAM Parity Check is supported. This error address is frozen from being updated until it is read by the VBUS host. Reading this register clears its contents to the default value The default value is 0x400 if Extended Buffer feature is enabled, else it is 0x200 Writes to this register are ignored

23.3.2.57 UERRADDR0 Register (Offset = 12Ch) [reset = 0h]

UERRADDR0 is shown in Figure 23-154 and described in Table 23-136.

Return to Summary Table.

Uncorrectable Parity or double bit ECC error address register - TXRAM

Figure 23-154. UERRADDR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																					UERRADDR0										
R-0h																					R-0h										

Table 23-136. UERRADDR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	UERRADDR0	R	0h	Uncorrectable Parity or double bit ECC error address This register holds the address when a parity error is generated while reading the MibSPI (Transmit) TXRAM. The TXRAM can be read either by CPU or by the MibSPI Sequencer FSM logic for transmission. The address captured is byte alligned. This error address is frozen from being updated until it is read by the VBUSP host. Reading this register clears its contents to the default value of 0x000. Writes to this register are ignored.

23.3.2.58 RXOVRN_BUF_ADDR Register (Offset = 130h) [reset = 200h]

RXOVRN_BUF_ADDR is shown in Figure 23-155 and described in Table 23-137.

Return to Summary Table.

Receive RAM Overrun Buffer Address Register

Figure 23-155. RXOVRN_BUF_ADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU					RXOVRN_BUF_ADDR										
R-0h					R-200h										

Table 23-137. RXOVRN_BUF_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	RXOVRN_BUF_ADDR	R	200h	Address of the RAM location of RXRAM for which an Overwrite occured. This address value will show only the offset address of the RAM location in the Multibuffer RAM address space. Refer to the device Spec for the actual absolute address of RXRAM. Content of this register are valid only when any of the TGINTVECT0 or TGINTVECT1 and SPIFLG registers show an RXOVRN error vector while in Multibuffer mode. If there are multiple Overrun errors, then this register holds address of first overrun address until it is read.

23.3.2.59 IOLPBKTSTCR Register (Offset = 134h) [reset = 0h]

IOLPBKTSTCR is shown in Figure 23-156 and described in Table 23-138.

Return to Summary Table.

SPI/MibSPI IO Loopback Test Control Register This register controls test mode for I/O pins. It also controls whether loop-back should be digital or analog ones in this test mode. In addition it contains control bits to induce some of the error condition into the module. These are to be used for test purpose only. All the control/status bits in this register are valid only when IO LPBK TST ENA field is set to “1010”.

Figure 23-156. IOLPBKTSTCR Register

31	30	29	28	27	26	25	24
NU4							SCSFAILFLG
R-0h							0h
23	22	21	20	19	18	17	16
NU3			CTRLBITERR	CTRLDESYNC	CTRLPARERR	CTRLTIMEOUT	CTRLDLENERR
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
NU2				IOLPBKTSTENA			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
NU1		ERRSCSPIN			CTRLSCSPINERR	LPBKTYPE	RXPENA
R-0h		R/W-0h			R/W-0h	R/W-0h	R/W-0h

Table 23-138. IOLPBKTSTCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
24	SCSFAILFLG		0h	Bit indicating a failure on SPISCS pin compare during analog loopback during IO Loopback Test mode. Read 1 = A comparison between the internal CSNR field and the analog looped back value of SPISCS[7:0] pins failed. A stuck-at fault is detected on one of the SPISCS[7:0]. Comparison is done only on the pins which are configured as functional and during transfer operation. 0 = No miscompares on any of the 8 chipselect pin value comparison with the internal Chipselect number CSNR during transfers. Write 1 = Clear this Flag bit. 0 = No effect.
23-21	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
20	CTRLBITERR	R/W	0h	Controls inducing of BITERR during IO Loopback Test mode. 1 = The value of incoming data from the loopback Transmit pin is flipped. 0 = No affect on BIT ERROR.
19	CTRLDESYNC	R/W	0h	Controls inducing of DESYNC Error during IO Loopback Test mode. 1 = Forces the incoming SPIENA pin (if functional) to remain ‘0’ even after the transfer complete. This forcing will be retained until the Kernel reaches IDLE state. 0 = No affect on DESYNC Error.
18	CTRLPARERR	R/W	0h	Controls inducing of Parity Error during IO Loopback Test mode. 1 = Flips the Parity Polarity signal being used for transmit parity generation logic 0 = No affect on Parity Error
17	CTRLTIMEOUT	R/W	0h	Controls inducing of TIMEOUT Error during IO LoopbacK Test mode. 1 = Forces the incoming SPIENA pin (if functional) to remain ‘1’ when transmission is initiated. The forcing will be retained until the Kernel reaches IDLE state. 0 = No affect on TIMEOUT Error.

Table 23-138. IOLPBKTSTCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CTRLDLENERR	R/W	0h	Controls inducing of Data Length Error during IO Loopback Test mode. 1 = When in Master mode, forces the SPIENA pin(if functional) to '1' when the module starts Shifting the data. When in Slave mode, forces the incoming SPISCS pin(if functional) to '1' when the module starts shifting the data.. 0 = No affect on Data Length Error.
15-12	NU2	R	0h	Reserved. Reads return '0' and writes have no effect.
11-8	IOLPBKTSTENA	R/W	0h	Module I/O Loopback Test Enable Key User and Privileged mode reads. Write access only in Privileged mode. Write: 1010 = I/O DFT is enabled All other values = I/O DFT is disabled Read: 1010 = I/O DFT is enabled All other values = I/O DFT is disabled
7-6	NU1	R	0h	Reserved. Reads return '0' and writes have no effect.
5-3	ERRSCSPIN	R/W	0h	Inject Error on ChipSelect Pin. The value in this field is decoded to find out the ChipSelect pin on which to inject an error. During the analog loopback of IO Loopback Test mode if CTRL SCS PIN ERR bit is set to '1', then the chipselect pin selected by this field is forced to the opposite of its original CSNR bit. 000 - Select SPISCS[0] for injecting error 001 - Select SPISCS[1] for injecting error . 111 - Select SPISCS[7] for injecting error
2	CTRLSCSPINERR	R/W	0h	Control bit to enable the injection of an error on SPISCS[7:0] pins. Individual pins of SPISCS[7:0] can be choosen using ERR SCS PIN. 1 = Enable the error inducing logic to the SPISCS pins. 0 = Disable the error inducing logic.
1	LPBKTYPE	R/W	0h	Module IO Loopback Type (Analog/Digital). User and Privileged mode reads. Write access only in Privileged mode. Write/Read : 1 = Analog loopback is enabled in module I/O DFT mode when IOLPBKTSTENA = 1010) 0 = Digital loopback is enabled in module I/O DFT mode when IOLPBKTSTENA = 1010
0	RXPENA	R/W	0h	Module Analog loopback through Receive Pin Enable. User and Privileged mode reads. Write only in privileged mode: Write/Read : 1 = Analog loopback through receive pin 0 = Analog loopback through transmit pin. This bit is valid only when LPBK TYPE = '1' which chooses Analog loopback mode.

23.3.2.60 EXTENDED_PRESCALE1 Register (Offset = 138h) [reset = 0h]

EXTENDED_PRESCALE1 is shown in Figure 23-157 and described in Table 23-139.

Return to Summary Table.

SPI/MibSPI Extended Prescale Register 1 (EXTENDED_PRESCALE1 for SPIFMT0 and SPIFMT1) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This is an extension of SPIFMT0 and SPIFMT1 registers. For example, EPRESCALE_FMT1(7:0) of EXTENDED_PRESCALE1 and PRESCALE1(7:0) of SPIFMT1 register will always reflect the same contents. Similarly EPRESCALE_FMT0(7:0) and PRESCALE0(7:0) of SPIFMT0 reflect the same contents.

Figure 23-157. EXTENDED_PRESCALE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2					EPRESCLAE_FMT1										
R-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1					EPRESCLAE_FMT0										
R-0h					R/W-0h										

Table 23-139. EXTENDED_PRESCALE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
26-16	EPRESCLAE_FMT1	R/W	0h	Extended Prescale value for SPIFMT1. EPRESCALE_FMT1 can be modified in privilege mode only. EPRESCALE_FMT1 determines the bit transfer rate of Data Format 1 if the SPI/MibSPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT1(7:0) bits and PRESCALE1(7:0) bits of SPIFMT1 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT1(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EXTENDED_PRESCALE1(26:16) or SPIFMT1(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE1 register is programmed after SPIFMT1 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE1 register when a prescale of more 256 is intended on SPICLK. BRFormatx = VBUSPCLK/(EXTENDEDPRESCALEy+1) When EPRESCALE_FMTy (y=1,2) is set to zero(0), the SPI clock rate defaults to VBUSPCLK/2.
15-11	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.

Table 23-139. EXTENDED_PRESCALE1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-0	EPRESCLAE_FMT0	R/W	0h	EPRESCALE_FMT0 can be modified in privilege mode only. EPRESCALE_FMT0 determines the bit transfer rate of Data Format 0 if the SPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT0(7:0) bits and PRESCALE0(7:0) bits of SPIFMT0 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT0(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EPRESCALE0(26:16) or SPIFMT0(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE1 register is programmed after SPIFMT0 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE1 register when a prescale of more 256 is intended on SPICLK.

23.3.2.61 EXTENDED_PRESCALE2 Register (Offset = 13Ch) [reset = 0h]

EXTENDED_PRESCALE2 is shown in Figure 23-158 and described in Table 23-140.

Return to Summary Table.

SPI/MibSPI Extended Prescale Register 2 (EXTENDED_PRESCALE2 for SPIFMT2 and SPIFMT3) This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This register is an extension of SPIFMT2 and SPIFMT3 registers. For example, EPRESCALE_FMT2(7:0) of EXTENDED_PRESCALE2 and PRESCALE2(7:0) of SPIFMT2 register will always reflect the same contents. Similarly EPRESCALE_FMT3(7:0) and PRESCALE3(7:0) of SPIFMT3 reflect the same contents.

Figure 23-158. EXTENDED_PRESCALE2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU4					EPRESCLAE_FMT3										
R-0h					R/W-0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3					EPRESCLAE_FMT2										
R-0h					R/W-0h										

Table 23-140. EXTENDED_PRESCALE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU4	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
26-16	EPRESCLAE_FMT3	R/W	0h	EPRESCALE_FMT3 can be modified in privilege mode only. EPRESCALE_FMT3 determines the bit transfer rate of Data Format 3 if the SPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT3(7:0) bits and PRESCALE3(7:0) bits of SPIFMT3 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT3(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EPRESCALE3(26:16) or SPIFMT3(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE2 register is programmed after SPIFMT3 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE2 register when a prescale of more 256 is intended on SPICLK.
15-11	NU3	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
10-0	EPRESCLAE_FMT2	R/W	0h	EPRESCALE_FMT2 can be modified in privilege mode only. EPRESCALE_FMT2 determines the bit transfer rate of Data Format 2 if the SPI is the network master. If the SPI / MibSPI is configured as slave, this field DOES NOT NEED to be configured. These EPRESCALE_FMT2(7:0) bits and PRESCALE2(7:0) bits of SPIFMT2 register will point to the same physically implemented register. Refer to Figure 56 for a graphical representation of the implementation. Write : This register field should be written if a SPICLK prescaler of more VBUSPCLK/256 is required. This field provides a prescaler of up to VBUSPCLK/2048 for SPICLK. Writing to this register field will also get reflected in SPIFMT3(15:8). Read : Reading this field will reflect the PRESCALE value based on the last written register field i.e., EXTENDED_PRESCALE2(26:16) or SPIFMT2(15:8) register. Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE2 register is programmed after SPIFMT2 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE2 register when a prescale of more 256 is intended on SPICLK.

23.3.2.62 ECCDIAG_CTRL Register (Offset = 140h) [reset = Ah]

ECCDIAG_CTRL is shown in Figure 23-159 and described in Table 23-141.

Return to Summary Table.

ECC Diagnostic Control register

Figure 23-159. ECCDIAG_CTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU												ECCDIAG_EN			
R-0h												R/W-Ah			

Table 23-141. ECCDIAG_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
3-0	ECCDIAG_EN	R/W	Ah	ECC Diagnostic mode Enable Key bits. 0101 : Diagnostic mode is enabled. Writes and reads from ECC bits allowed from the ECC address space. Refer to Section 9 for details on ECC/Parity address space. Others : Diagnostic mode is disabled. No writes to ECC bits are ignored, reads return ‘0’.

23.3.2.63 ECCDIAG_STAT Register (Offset = 144h) [reset = 0h]

ECCDIAG_STAT is shown in Figure 23-160 and described in Table 23-142.

Return to Summary Table.

ECC Diagnostic Status register

Figure 23-160. ECCDIAG_STAT Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2						DEFLG1	DEFLG0
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
NU1							
R-0h							
7	6	5	4	3	2	1	0
NU1						SEFLG1	SEFLG0
R-0h						R-0h	R-0h

Table 23-142. ECCDIAG_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	NU2	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
17	DEFLG1	R	0h	Double bit error flag for RXRAM 1 - A double bit Error is detected for RXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.
16	DEFLG0	R	0h	Double bit error flag for TXRAM 1 - A double bit Error is detected for TXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.
15-2	NU1	R	0h	Reserved. Reads return ‘0’ and writes have no effect.
1	SEFLG1	R	0h	Single bit error flag for RXRAM 1 - A Single bit Error is detected for RXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.
0	SEFLG0	R	0h	Single bit error flag for TXRAM 1 - A Single bit Error is detected for TXRAM bank during diagnostic mode tests. 0 - No error. A write ‘1’ to this bit will clear the bit.

23.3.2.64 SBERRADDR1 Register (Offset = 148h) [reset = 0h]

SBERRADDR1 is shown in Figure 23-161 and described in Table 23-143.

Return to Summary Table.

Single Bit Error Address Register - RXRAM

Figure 23-161. SBERRADDR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU1																					SBERRADDR1										
R-0h																					R-0h										

Table 23-143. SBERRADDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU1	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	SBERRADDR1	R	0h	Single Bit ECC Error Address This register holds the address of the RAM location when a single bit error is generated by SECDED block while reading the MibSPI (Receive) RXRAM. This error address is frozen from being updated until it is read by the VBUS host. Reading this register clears its contents to the default value The default value is 0x400 if Extended Buffer feature is enabled, else it is 0x200 Writes to this register are ignored.

23.3.2.65 SBERRADDR0 Register (Offset = 14Ch) [reset = 0h]

SBERRADDR0 is shown in Figure 23-162 and described in Table 23-144.

Return to Summary Table.

Single Bit ECC Error Address Register - TXRAM

Figure 23-162. SBERRADDR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2																					SBERRADDR0										
R-0h																					R-0h										

Table 23-144. SBERRADDR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	NU2	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
10-0	SBERRADDR0	R	0h	Single Bit ECC Error Address This register holds the address when a single bit error is generated from SECDED block while reading the MibSPI (Transmit) TXRAM. The TXRAM can be read either by CPU or by the MibSPI Sequencer logic for transmission. This error address is frozen from being updated until it is read by the VBUSP host. Reading this register clears its contents to the default value of 0x000. Writes to this register are ignored.

23.3.2.66 SPIREV Register (Offset = 1FCh) [reset = 4A050308h]

SPIREV is shown in Figure 23-163 and described in Table 23-145.

Return to Summary Table.

SPI / MibSPI Revision ID Register

Figure 23-163. SPIREV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		NU		FUNC											
R-1h		R-0h		R-A05h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-0h					R-3h			R-0h		R-8h					

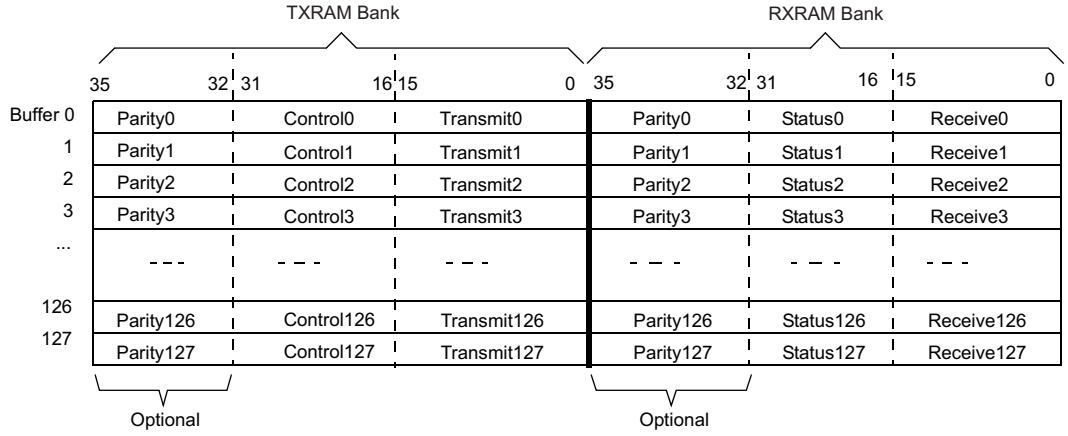
Table 23-145. SPIREV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Identification Scheme Used to distinguish different ID schemes. Reads 0x01
29-28	NU	R	0h	Reserved.Reads return ‘0’ and writes have no effect.
27-16	FUNC	R	A05h	Indicates functionally equivalent module family Reads 0xA05
15-11	RTL	R	0h	RTL version number Read value will provide an approximate RTL revision number. The design release version can be obtained from the device specification
10-8	MAJOR	R	3h	Major Revision number Reads 0x3
7-6	CUSTOM	R	0h	Indicates device specific implementation Reads 0x0
5-0	MINOR	R	8h	Minor Revision number Reads 0x8

23.4 Multi-Buffer RAM

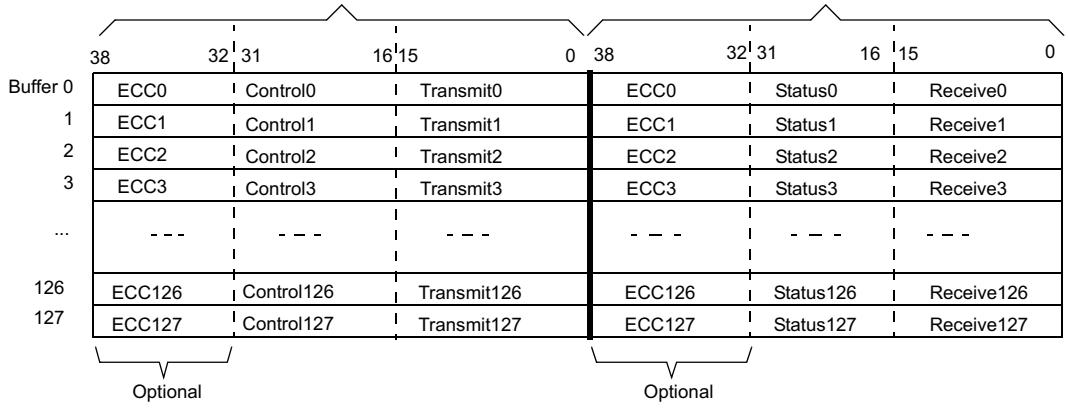
The multi-buffer RAM comprises of all buffers, which can be configured identically. The multi-buffer RAM contains two banks of up to128/256 words of 32 bits for a maximum configuration, one each for TXRAM (replicating the SPIDAT1 register) and RXRAM (replicating the SPIBUF register). The buffers can be partitioned into multiple transfer groups, each containing a variable number of buffers. Each of the buffers can be sub-divided into a 16-bit transmit field, a 16-bit receive field, a 16-bit control field, and a 16-bit status field. A 4-bit parity field per word is also included in each RAM bank, as shown in Figure 23-164. If ECC support is implemented for RAM fault detection, then a 7-bit ECC field per word is also included in each RAM bank, as shown in Figure 23-165.

Figure 23-164. Multi-Buffer RAM Configuration When Parity Check is Supported



Depth will be up to 256 buffers, if EXTENDED_BUF feature is implemented.

Figure 23-165. Multi-Buffer RAM Configuration When ECC Check is Supported



Depth will be up to 256 buffers, if EXTENDED_BUF feature is implemented.

All fields can be read and written with 8-bit, 16-bit, or 32-bit accesses.

The transmit fields can be written and read in the address range 000h to 1FFh. The transmit words contain data and control fields.

The receive RAM fields are read-only and can be accessed through the address range 200h to 3FCh. The receive words contain data and status fields.

The chip select number bit field CSNR[7:0] of the control field for a given word is mirrored into the corresponding receive-buffer status field after transmission.

The Parity is automatically calculated and copied to Parity location

NOTE: Refer to the specific device datasheet for the actual number of transmit and receive buffers.

Write to unimplemented buffer is overwriting the corresponding implemented buffer. In MIBSPI, if the RAM SIZE specified is 32 buffers, write to 33rd buffer overwrites 1st buffer, write to 34th buffer overwrites 2nd buffer, and so on.

23.4.1 Multi-Buffer RAM Auto Initialization

When the MIBSPI is out of reset mode, auto initialization of multi-buffer RAM starts. The application code must check for BUFINITACTIVE bit to be 0 (multi-buffer RAM initialization is complete) before configuring multi-buffer RAM.

Besides the default auto initialization after reset, the auto-initialization sequence can also be done in following ways:

1. Enable the global hardware memory initialization key by programming a value of 1010b to the bits [3:0] of the MINITGCR register of the System module.
2. Set the control bit for the multi-buffer RAM in the MSINENA System module register. This bit is device-specific for each memory that support auto-initialization. Please refer to the device datasheet to identify the control bit for the multi-buffer RAM. This starts the initialization process. The BUFINITACTIVE bit will get set to reflect that the initialization is ongoing.
3. When the memory initialization is completed, the corresponding status bit in the MINISTAT register will be set. Also, the BUFINITACTIVE bit will get cleared.
4. Disable the global hardware memory initialization key by programming a value of 0101 to the bits [3:0] of the MINITGCR register of the System module.

Please refer to the Architecture User Guide for more details on the memory auto-initialization process.

NOTE: During Auto Initialization process, all the multi-buffer mode registers (except MIBSPIE) will be reset to their default values. So, it should be ensured that Auto Initialization is completed before configuring the multi-buffer mode register.

23.4.2 Multi-Buffer RAM Register Summary

This section describes the multi-buffer RAM control and transmit-data fields of each word of TXRAM, and the status and receive-data fields of each word of RXRAM. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

Table 23-146. Multi-Buffer RAM Register

Offset	Acronym	Register Description	Section
Base + 0h-1FFh	TXRAM	Multi-Buffer RAM Transmit Data Register	Section 23.4.3
Base + 200h-3FFh	RXRAM	Multi-Buffer RAM Receive Buffer Register	Section 23.4.4

23.4.3 Multi-Buffer RAM Transmit Data Register

Each word of TXRAM is a transmit-buffer register.

Figure 23-166. Multi-Buffer RAM Transmit Data Register [offset = Base + 000-1FFh]

31 29	28	27	26	25 24	23 16
BUFMODE	CSHOLD	LOCK	WDEL	DFSEL	CSNR
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15 0
TXDATA
R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-147. Multi-buffer RAM Transmit Data Register Field Descriptions

Bit	Field	Value	Description
31-29	BUFMODE		Specify conditions that are recognized by the sequencer to initiate transfers of each buffer word. When one of the "skip" modes is selected, the sequencer checks the buffer status every time it reads from this buffer. If the current buffer status (TXFULL, RXEMPTY) does not match, the buffer is skipped without a data transfer. When one of the "suspend" modes is selected, the sequencer checks the buffer status when it reads from this buffer. If TXFULL and/or RXEMPTY do not match, the sequencer waits until a match occurs. No data transfer is initiated until the status condition of this buffer changes.
		0	**disabled.** The buffer is disabled
		1h	**skip single-transfer mode.** Skip this buffer until the corresponding TXFULL flag is set (new transmit data is available).
		2h	**skip overwrite-protect mode.** Skip this buffer until the corresponding RXEMPTY flag is set (new receive data can be stored in RXDATA without data loss).
		3h	**skip single-transfer overwrite-protect mode**. Skip this buffer until both of the corresponding TXFULL and RXEMPTY flags are set. (new transmit data available and previous data received by the host).
		4h	**continuous mode.** Initiate a transfer each time the sequencer checks this buffer. Data words are retransmitted if the buffer has not been updated. Receive data is overwritten, even if it has not been read.
		5h	**suspend single-transfer mode.** Suspend-to-wait until the corresponding TXFULL flag is set (the sequencer stops at the current buffer until new transmit data is written in the TXDATA field).
		6h	**suspend overwrite-protect mode.** Suspend-to-wait until the corresponding RXEMPTY flag is set (the sequencer stops at the current buffer until the previously-received data is read by the host.
		7h	**suspend single-transfer overwrite-protect mode.** Suspend-to-wait until the corresponding TXFULL and RXEMPTY flags are set (the sequencer stops at the current buffer until new transmit data is written into the TXDATA field and the previously-received data is read by the host).
28	CSHOLD		Chip select hold mode. The CSHOLD bit is supported in master mode only in compatibility-mode of MibSPI, (it is ignored in slave mode). CSHOLD defines the behavior of the chip select line at the end of a data transfer.
		0	The chip select signal is deactivated at the end of a transfer after the T2CDELAY time has passed. If two consecutive transfers are dedicated to the same chip select this chip select signal will be deactivated for at least 2VCLK cycles before it is activated again.
		1	The chip select signal is held active at the end of a transfer until a control field with new data and control information is loaded into SPIDAT1. If the new chip select number equals the previous one, the active chip select signal is extended until the end of transfer with CSHOLD cleared, or until the chip-select number changes.
27	LOCK		Lock two consecutive buffer words. Do not allow interruption by TG's with higher priority.
		0	Any higher-priority TG can begin at the end of the current transaction.
		1	A higher-priority TG cannot occur until after the next unlocked buffer word is transferred.

Table 23-147. Multi-buffer RAM Transmit Data Register Field Descriptions (continued)

Bit	Field	Value	Description
26	WDEL		Enable the delay counter at the end of the current transaction.
			Note: The WDEL bit is supported in master mode only. In slave mode, this bit will be ignored.
		0	No delay will be inserted. However, SPISCS pins will still be de-activated for at least for 2VCLK cycles if CSHOLD = 0.
			Note: The duration for which the SPISCS pin remains deactivated also depends upon the time taken to supply a new word after completing the shift operation (in compatibility mode). If TXBUF is already full, then the SPISCS will be deasserted for at least two VCLK cycles (if WDEL = 0).
		1	After a transaction, WDELAY of the corresponding data format will be loaded into the delay counter. No transaction will be performed until the WDELAY counter overflows. The SPISCS pins will be de-activated for at least (WDELAY + 2) * VCLK_Period duration.
25-24	DFSEL		Data word format select
		0	Data word format 0 is selected
		1h	Data word format 1 is selected
		2h	Data word format 2 is selected
		3h	Data word format 3 is selected
23-16	CSNR	0-FFh	Chip select number. CSNR defines the chip-select that will be activated during the data transfer.
			Note: Writing to only the control field (using byte writes) does not initiate any SPI transfer in master mode. This feature can be used to set up SPICLK phase or polarity before actually starting the transfer by only updating the DFSEL fields in the control field to select the required phase/polarity combination.
15-0	TXDATA	0-7FFFh	**Transfer data.**When written, these bits are copied to the shift register if it is empty. If the shift register is not empty, then they are held in TXBUF.
			SPIEN must be set to 1 before this register can be written to. Writing a 0 to SPIEN forces the lower 16 bits of SPIDAT1 to 0.
			Write to this register ONLY when using the automatic slave chip-select feature (see Section 23.2.1.1 for more information). A write to this register will drive the contents of CSNR[7:0] on the SPISCS[3:0] pins, if they are configured as functional pins.
			When this register is read, the contents of TXBUF, which holds the latest data written, will be returned.
			Note: Regardless of the character length, the transmit data should be right-justified before writing to the SPIDAT1 register.

23.4.4 Multi-buffer RAM Receive Buffer Register

Each word of RXRAM is a receive-buffer register.

Figure 23-167. Multi-buffer RAM Receive Buffer Register [offset = RAM Base + 200-3FFh]

31	30	29	28	27	26	25	24
RXEMPTY	RXOVR	TXFULL	BITERR	DESYNC	PARITYERR	TIMEOUT	DLENERR
RS-1	RC-0	R-0	RC-0	RC-0	RC-0	RC-0	RC-0
23							16
LCSNR							
R-0							
15							0
RXDATA							
R/W-0							

LEGEND: R = Read only; R/W = Read/Write; C = Clear; S = Set; -*n* = value after reset

Table 23-148. Multi-buffer Receive Buffer Register Field Descriptions

Bit	Field	Value	Description
31	RXEMPTY		**Receive data buffer empty.**When the host reads the SPIBUF field or the entire SPIBUF register, it automatically sets the RXEMPTY flag. When a data transfer is completed, the received data is copied into SPIBUF, and the RXEMPTY flag is cleared.
		0	New data has been received and copied into the SPIBUF field.
		1	No data has been received since the last read of SPIBUF.
			This flag gets set to 1 under the following conditions: • Reading the RXDATA portion of the SPIBUF register. • Writing a 1 to clear the RXINTFLG bit in the SPI Flag Register (SPIFLG). Write-clearing the RXINTFLG bit before reading the SPIBUF indicates the received data is being ignored. Conversely, RXINTFLG can be cleared by reading the RXDATA portion of SPIBUF (or the entire register).
30	RXOVR		Receive data buffer overrun. When a data transfer is completed and the received data is copied into RXBUF while it is already full, RXOVR is set. Overruns always occur to RXBUF, not to SPIBUF; the contents of SPIBUF are overwritten only after it is read by the Peripheral (VBUSP) master (CPU, DMA, or other host processor). If enabled, the RXOVRN interrupt is generated when RXBUF is overwritten, and reading either SPI Flag Register (SPIFLG) or SPIVECTx shows the RXOVRN condition. Two read operations from the SPIBUF register are required to reach the overwritten buffer word (one to read SPIBUF, which then transfers RXDATA into SPIBUF for the second read). This flag is cleared to 0 when the RXDATA is read. **Note: A special condition under which RXOVR flag gets set.If both SPIBUF and RXBUF are already full and while another buffer receive is underway, if any errors such as TIMEOUT, BITERR and DLEN_ERR occur, then RXOVR in RXBUF and SPI Flag Register (SPIFLG) will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a normal receive overrun.**
		0	No receive data overrun condition occurred since last read of the data field.
		1	A receive data overrun condition occurred since last read of the data field.
29	TXFULL		**Transmit data buffer full.**This flag is a read-only flag. Writing into the SPIDAT0 or SPIDAT1 field while the TX shift register is full will automatically set the TXFULL flag. Once the word is copied to the shift register, the TXFULL flag will be cleared. Writing to SPIDAT0 or SPIDAT1 when both TXBUF and the TX shift register are empty does not set the TXFULL flag.
		0	The transmit buffer is empty; SPIDAT0/SPIDAT1 is ready to accept a new data.
		1	The transmit buffer is full; SPIDAT0/SPIDAT1 is not ready to accept new data.

Table 23-148. Multi-buffer Receive Buffer Register Field Descriptions (continued)

Bit	Field	Value	Description
28	BITERR		**Bit error.**There was a mismatch of internal transmit data and transmitted data.
		0	No bit error occured.
			Note: This flag is cleared to 0 when the RXDATA portion of the SPIBUF register is read.
		1	A bit error occurred. The SPI samples the signal of the transmit pins (master: SIMOx, slave: SOMIx) at the receive point (one-half clock cycle after the transmit point). If the sampled value differs from the transmitted value, a bit error is detected and the BITERR flag is set. Possible reasons for a bit error include noise, an excessively high bit rate, capacitive load, or another master/slave trying to transmit at the same time.
27	DESYNC		**Desynchronization of slave device.**This bit is valid in master mode only.
			The master monitors the ENA signal coming from the slave device and sets the DESYNC flag if ENA is deactivated before the last reception point or after the last bit is transmitted plus $t_{T2EDELAY}$. If DESYNCENA is set, an interrupt is asserted. Desynchronization can occur if a slave device misses a clock edge coming from the master.
			Note: In the Compatibility Mode MibSPI, under some circumstances it is possible for a desync error detected for the previous buffer to be visible in the current buffer. This is because the receive completion flag/interrupt is generated when the buffer transfer is completed. But desynchronization is detected after the buffer transfer is completed. So, if the VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct desync condition. In multi-buffer mode, the desync flag is always guaranteed to be for the current buffer.
		0	No slave desynchronization detected.
			Note: This flag is cleared to 0 when the RXDATA portion of the SPIBUF register is read.
		1	A slave device is desynchronized.
26	PARITYERR		**Parity error.**The calculated parity differs from the received parity bit.
			If the parity generator is enabled (selected individually for each buffer) an even or odd parity bit is added at the end of a data word. During reception of the data word, the parity generator calculates the reference parity and compares it to the received parity bit. If a mismatch is detected, the PARITYERR flag is set.
			Note: This flag is cleared to 0 when the RXDATA portion of the SPIBUF register is read.
		0	No parity error detected.
		1	A parity error occured.
25	TIMEOUT		Time-out because of non-activation of ENA pin.
			The SPI generates a time-out when the slave does not respond in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected, the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition, the TIMEOUT flag in the status field of the corresponding buffer and in the SPI Flag Register (SPIFLG) is set.
			This bit is valid only in master mode.
			This flag is cleared to 0 when RXDATA portion of the SPIBUF register is read.
		0	No ENA-pin time-out occured.
		1	An ENA signal time-out occured.
24	DLENERR		**Data length error flag.**
			Note: This flag is cleared to 0 when the RXDATA portion of the SPIBUF register is read.
		0	No data-length error has occurred.
		1	A data length error has occurred.
23-16	LCSNR	0-FFh	**Last chip select number.** LCSNR in the status field is a copy of CSNR in the corresponding control field. It contains the chip select number that was activated during the last word transfer.
15-0	RXDATA	0-FFFFh	**SPI receive data.** This is the received word, transferred from the receive shift-register at the end of a transfer. Regardless of the programmed character length and the direction of shifting, the received data is stored right-justified in the register.

23.5 Parity\ECC Memory

Parity/ECC portion of multi-buffer RAM is not accessible by the CPU during normal operating modes. However each read or write operation to the Control/Data/Status portion of the multi-buffer RAM causes reads/writes to the parity/ECC portion as well.

- Each write to the multi-buffer RAM (either from the VBUS interface or by the MibSPI itself) causes a write operation to the Parity/ECC portion of RAM simultaneously to update the equivalent parity/ECC bits.
- Each read operation from the multi-buffer RAM (either from the VBUS interface or by the MibSPI itself) causes a read operation from the Parity/ECC portion of the RAM for parity/ECC comparison purpose.
- Reads/Writes to multi-buffer RAM could either be caused by any CPU/DMA accesses or by the Sequencer logic of MibSPI itself.

For testing the Parity/ECC portion of the multi-buffer RAM which is a 4bit or 7bit field per word address, a separate Parity/ECC Memory Test mode is available. Parity Memory Test Mode can be selecting the PTESTEN bit in PAR_ECC_CTRL register in addition to PAR_ECC_CTRL.PTESTEN bit, ECCDIAG_REG(3:0) should be enabled

During the parity test mode, the parity locations are addressable at the address between RAM_BASE_ADDR + 0x400h and RAM_BASE_ADDR + 0x7FFh. Each location corresponds, sequentially, to each TXRAM word, then to each RXRAM word. See Figure 23-168 for a diagram of the memory map of parity memory during normal operating mode and during parity test mode while EXTENDED_BUF mode is disabled or the feature is not implemented. See Figure 23-169 for a diagram of the memory map of parity memory during normal operating mode and during parity test mode while EXTENDED_BUF mode is enabled.

During Parity/ECC test mode, after writing the Data/Control portion of the RAM, the Parity/ECC locations can be written with wrong parity/ECC bits to intentionally cause Parity/ECC Errors.

See the device-specific data sheet to get the actual base address of the multi-buffer RAM.

NOTE: The RX_RAM_ACCESS bit can also be set to 1 during the Parity/ECC Test mode to be able to write to RXRAM locations for test purpose. Both Parity/ECC bits testing and RXRAM testing can be done together.

There are 4 bits of parity corresponding to each of the 32-bit multi-buffer locations. Individual bits in the parity memory are byte-addressable in parity test mode. See the example in Section 23.5.1 for further details.

If ECC is enabled, there are 7 ECC-bits corresponding to each of the 32-bit multi-buffer locations. See the example in Section 23.5.1 for further details.

NOTE: Polarity of the parity (odd/even) varies by device. In some devices, a control register in the system module can be used to select odd or even parity.

Figure 23-168. Memory-Map for Parity Locations During Normal and Test Mode While EXTENDED_BUF Mode is Disabled or the Feature is Not Implemented

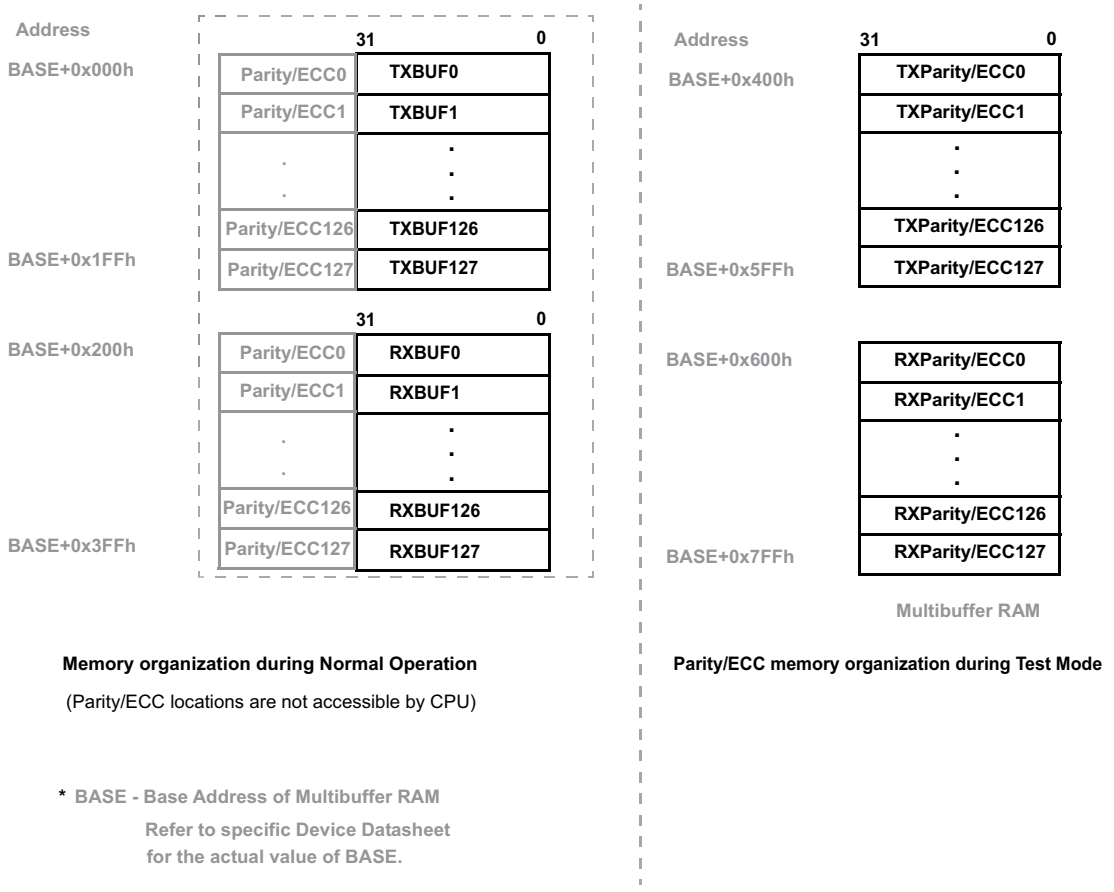
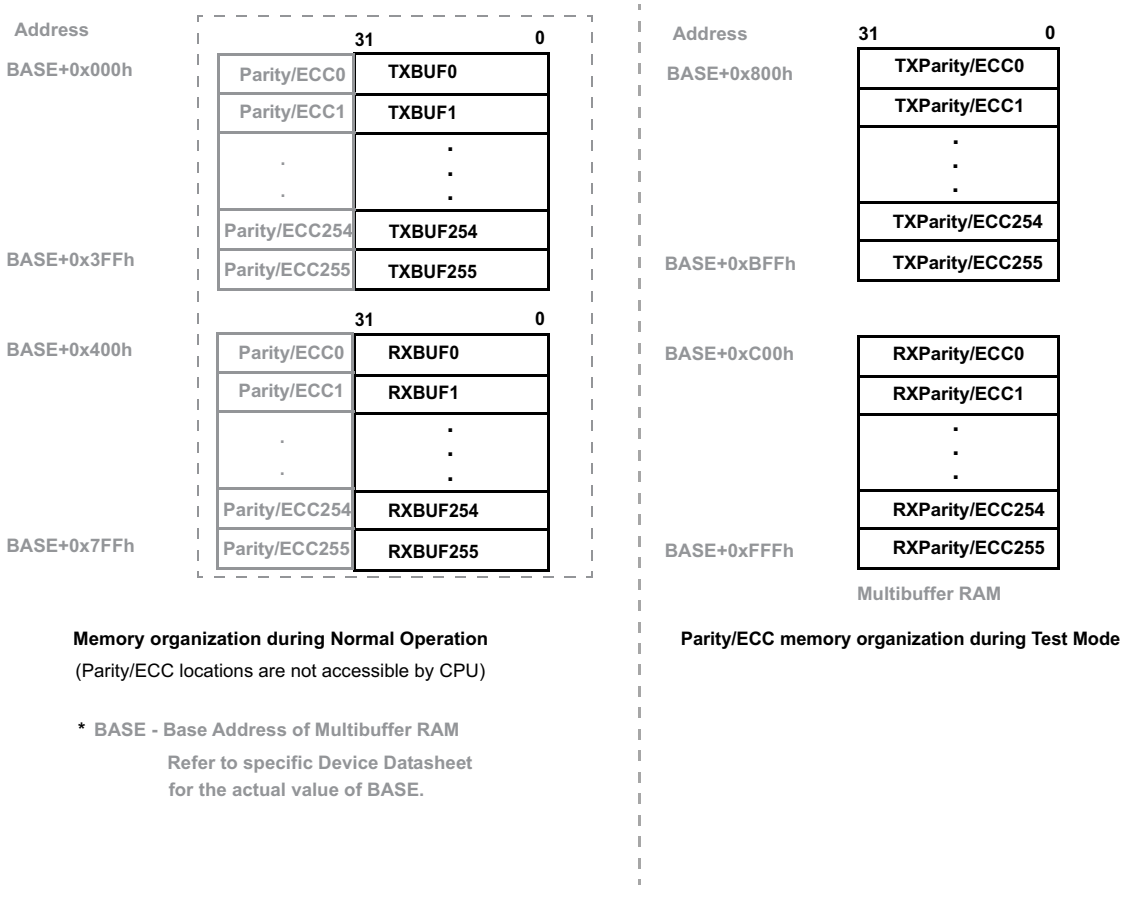


Figure 23-169. Memory-Map for Parity Locations During Normal and Test Mode While EXTENDED_BUF Mode is Enabled

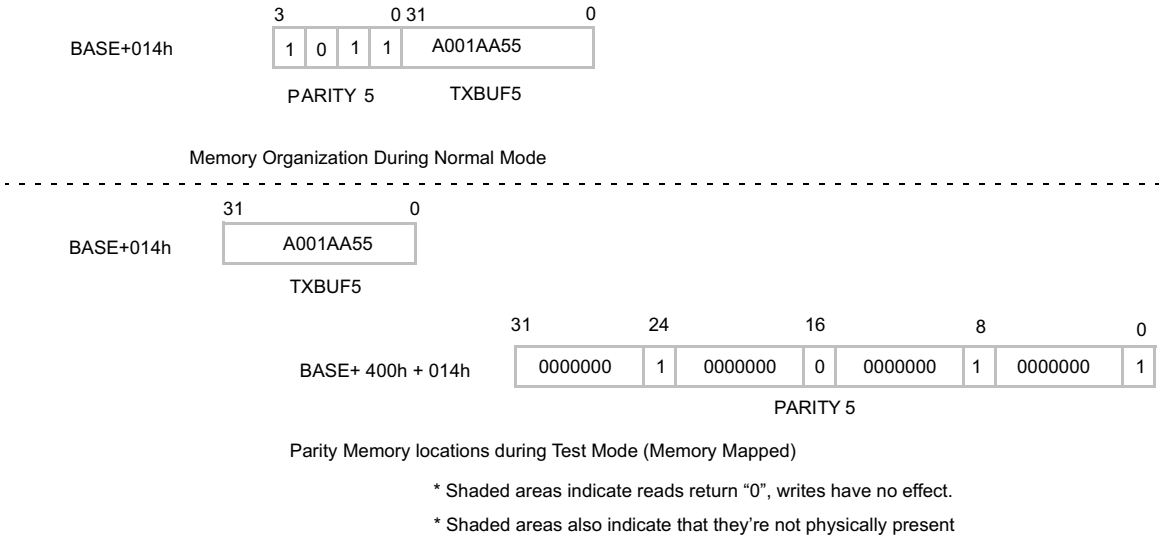


23.5.1 Example of Parity Memory Organization

Suppose TXBUF5 (6th location in TXRAM) in the multi-buffer RAM is written with a value of A001_AA55. If the polarity of the parity is set to odd, the corresponding parity location parity5 will get updated with equivalent parity of 1011 in its field.

During parity-memory test mode, these bits can be individually byte addressed. The return data will be a byte adjusted with actual parity bit in the LSB of the byte. If a word is read from the word-boundary address of parity locations, then each bit of the 4-bit parity is byte-adjusted and a 32-bit word is returned. 0s will be padded into the parity bits to get each byte. See Figure 23-170 for a diagram.

Figure 23-170. Example of Memory-Mapped Parity Locations During Test Mode



NOTE: Read Access to Parity Memory Locations

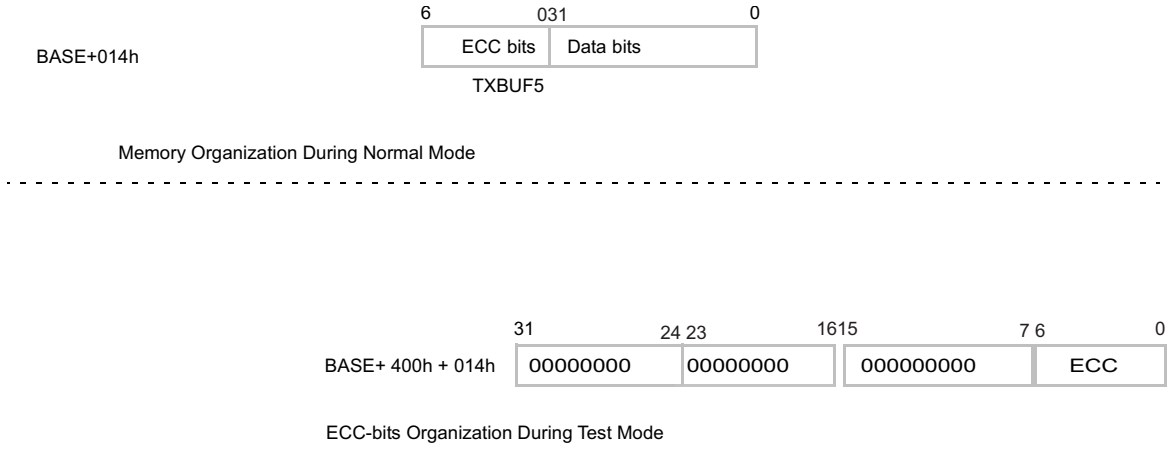
Parity memory locations can be read even without entering into parity memory test mode. Their address remains as in memory test mode. It is only to enter parity-memory test mode to enable write access to the parity memory locations.

23.5.2 Example of ECC Memory Organization

Suppose TXBUF5 (6th location in TXRAM portion) in the multi-buffer RAM is written with a value of A001_AA55, then the corresponding ECC-bits will be updated in ECC location.

The ECC bits can be accessed by user, when Memory Test mode is enabled and additionally diagnostic mode is also enabled. The actual ECC bits will be aligned as shown in Figure 23-171.

Figure 23-171. Example of ECC Bit Locations During Test Mode



NOTE: Access to ECC locations

ECC locations can be read/write only when Parity Memory Test mode and diagnostic mode is enabled

23.6 MibSPI Pin Timing Parameters

The pin timings of SPI can be classified based on its mode of operation. In each mode, different configurations like Phase & Polarity affect the pin timings.

The pin directions are based on the mode of operation.

Master mode SPI:

- SPICLK (SPI Clock) - Output
- SPISIMO (SPI Slave In Master Out) - Output
- $\overline{\text{SPISCS}}$[7:0] (SPI Slave Chip Selects) - Output
- SPISOMI (SPI Slave Out Master In) - Input
- $\overline{\text{SPIENA}}$ (SPI slave ready Enable) - Input

Slave mode SPI:

- SPICLK - Input
- SPISIMO - Input
- $\overline{\text{SPISCS}}$ - Input
- SPISOMI - Output
- $\overline{\text{SPIENA}}$ - Output

NOTE: All the following timing diagrams are with Phase = 0 and Polarity = 0, unless explicitly stated otherwise.

23.6.1 Master Mode Timings for SPI/MibSPI

Figure 23-172. SPI/MibSPI Pins During Master Mode 3-Pin Configuration

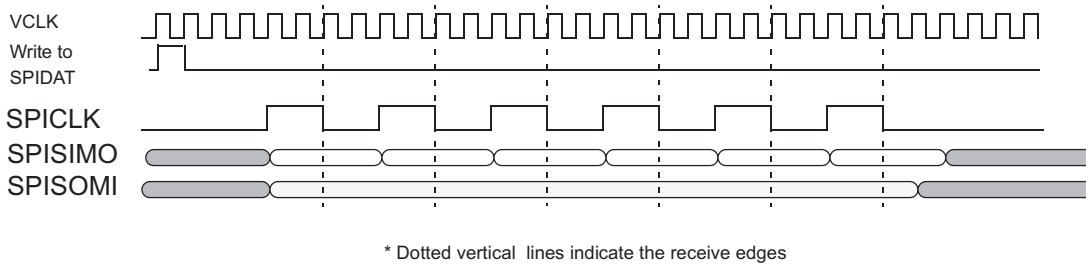


Figure 23-173. SPI/MibSPI Pins During Master Mode 4-Pin with $\overline{\text{SPISCS}}$ Configuation

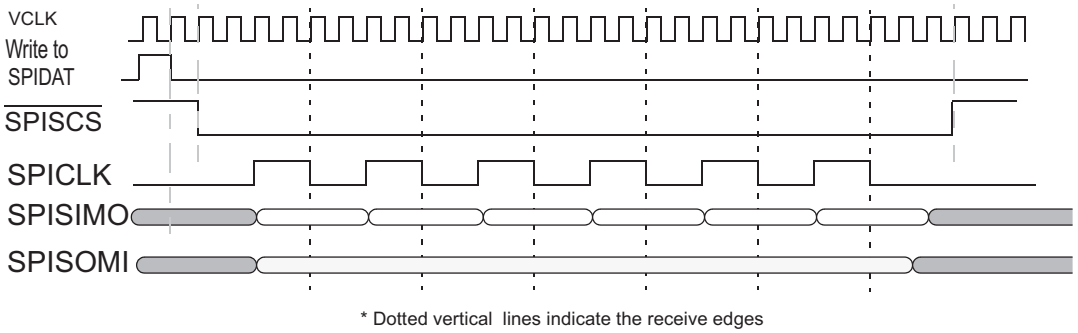
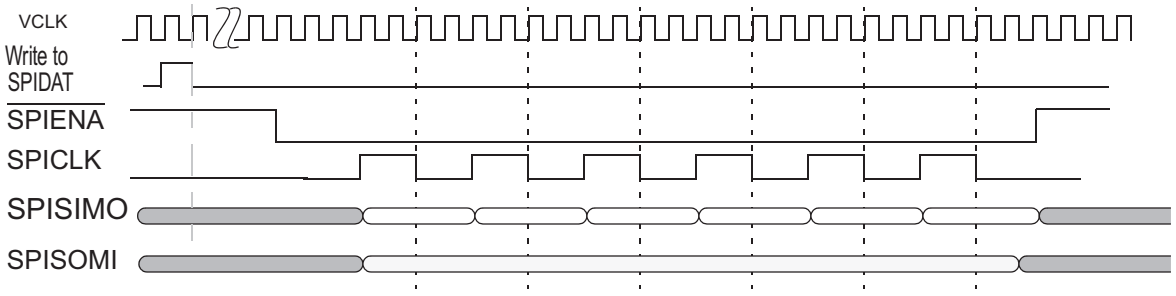
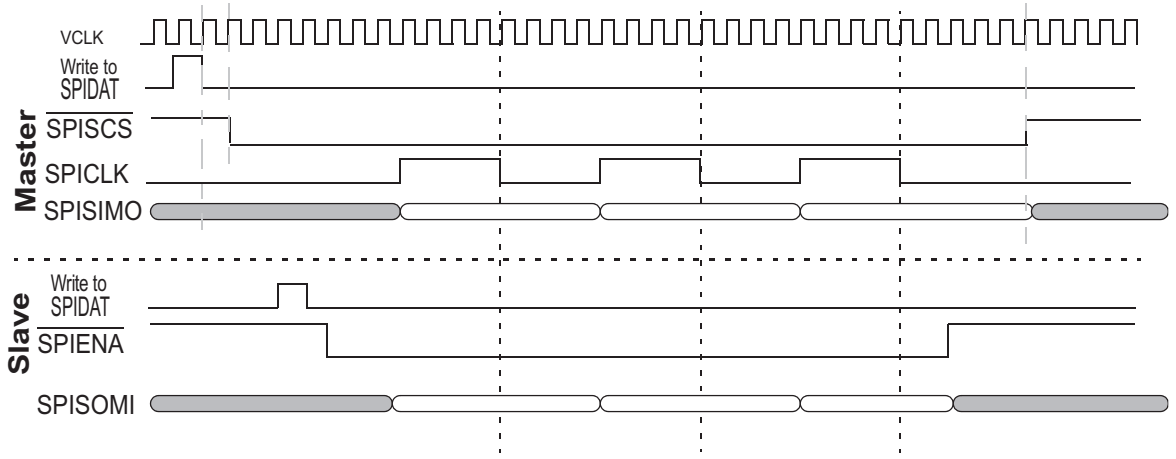


Figure 23-174. SPI/MibSPI Pins During Master Mode in 4-Pin with $\overline{\text{SPIENA}}$ Configuration



* De-activation of $\overline{\text{SPIENA}}$ pin is controlled by the Slave.
* Dotted vertical lines indicate the receive edges

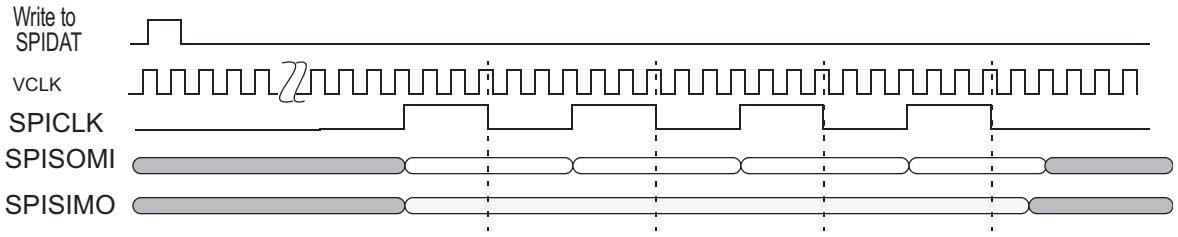
Figure 23-175. SPI/MibSPI Pins During Master/Slave Mode with 5-Pin Configuration



* Dotted vertical lines indicate the receive edges for the Master
* ENABLE_HIGHZ is set to '0' in Slave SPI

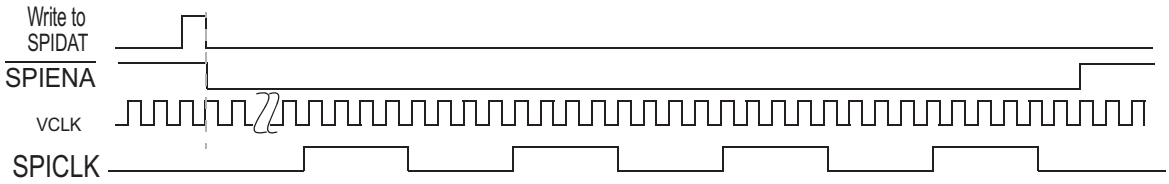
23.6.2 Slave Mode Timings for SPI/MibSPI

Figure 23-176. SPI/MibSPI Pins During Slave Mode 3-Pin Configuration



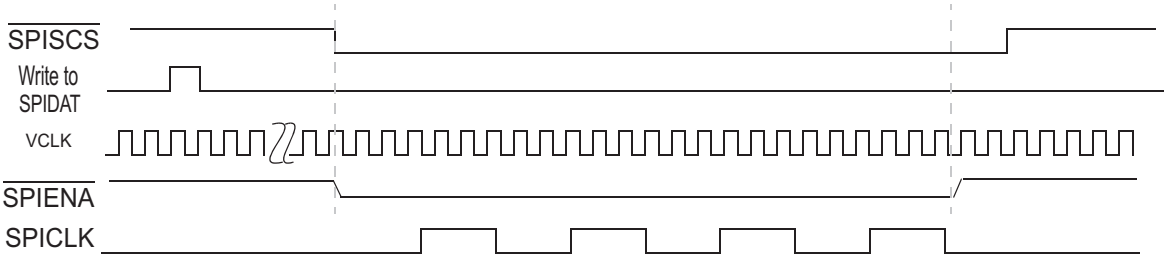
* Dotted vertical lines indicate the receive edges

Figure 23-177. SPI/MibSPI Pins During Slave Mode in 4-Pin with $\overline{\text{SPIENA}}$ Configuration



* Diagram shows a relationship between the $\overline{\text{SPIENA}}$ from Slave and SPICLK from Master

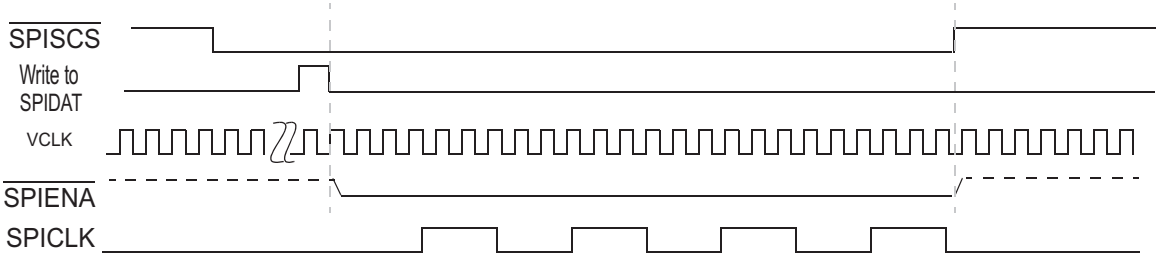
Figure 23-178. SPI/MibSPI Pins During Slave Mode in 5-Pin Configuration (Single Slave)



* ENABLE_HIGHZ is set to '0' in Slave SPI

* Diagram shows relationship between the $\overline{\text{SPISCS}}$ from a Master to $\overline{\text{SPIENA}}$ from Slave SPI when $\overline{\text{SPIENA}}$ is configured in Push-Pull mode

Figure 23-179. SPI/MibSPI Pins During Slave Mode in 5-Pin Configuration (Single/Multi-Slave)



* ENABLE_HIGHZ is set to '1' in Slave SPI

* Diagram shows relationship between the $\overline{\text{SPISCS}}$ from a Master to $\overline{\text{SPIENA}}$ from Slave SPI when $\overline{\text{SPIENA}}$ is configured in High-Z mode

23.6.3 Master Mode Timing Parameter Details

In case of Master, the module drives out SPICLK. It also drives out the Transmit data on SPISIMO with respect to its internal SPICLK. In case of Master mode, the RX data on SPISOMI pin is registered with respect to SPICLK recieved through Input buffer from the I/O pad.

If Chip Select pin is functional, then Master will drive out $\overline{\text{SPISCS}}$ pin(s) before starting the SPICLK. If $\overline{\text{SPIENA}}$ pin is functional, then Master will wait for an active 'low' from Slave on the input pin to start the SPICLK.

23.6.4 Slave Mode Timing Parameter Details

In case of Slave mode, the module will drive only SPISOMI and SPIENA pins. All other pins are inputs to it. The RX data on SPISIMO pin will be registered with respect to SPICLK pin. Slave will use $\overline{\text{SPISCS}}$ pin to drive out $\overline{\text{SPIENA}}$ pin if both are functional. If 4pin with $\overline{\text{SPIENA}}$ is configured, then Slave will drive out active low signal on $\overline{\text{SPIENA}}$ pin when a new data is written to the TX Shift Register. Irrespective of 4pin with $\overline{\text{SPIENA}}$ or 5pin configuration, the Slave will deassert the $\overline{\text{SPIENA}}$ pin after the last bit is received. If ENABLE_HIGHZ (SPIINT0.24) bit is 0, the de-asserted value of $\overline{\text{SPIENA}}$ pin will be 1. Otherwise, it will depend upon the internal PullUp or PullDown resistor (if implemented) depending upon the Specification of the Chip.

Quad Serial Peripheral Interface (QSPI)

Topic **Page**

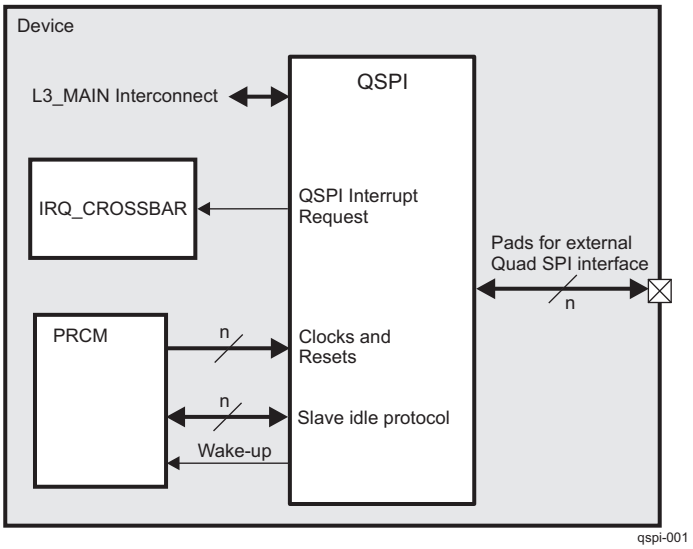
24.1 Quad Serial Peripheral Interface Overview 3158
24.2 QSPI Functional Description 3159
24.3 QSPI Register Manual 3167

24.1 Quad Serial Peripheral Interface Overview

The quad serial peripheral interface (QSPI™) module is a kind of SPI module that allows single, dual, or quad read access to external SPI devices. This module has a memory mapped register interface, which provides a direct interface for accessing data from external SPI devices and thus simplifying software requirements. The QSPI works as a master only.

The one QSPI in the device is primarily intended for fast booting from quad-SPI flash memories. Figure 24-1 shows the QSPI module overview.

Figure 24-1. QSPI Overview



The QSPI supports the following features:

- General SPI features:
  - Programmable clock divider
  - Six pin interface
  - Programmable length (from 1 to 128 bits) of the words transferred
  - Programmable number (from 1 to 4096) of the words transferred
  - 4 external chip-select signals
  - Support for 3-, 4-, or 6-pin SPI interface
  - Optional interrupt generation on word or frame (number of words) completion
  - Programmable delay between chip select activation and output data from 0 to 3 QSPI clock cycles
  - Programmable signal polarities
  - Programmable active clock edge
  - Software-controllable interface allowing for any type of SPI transfer
  - Control through L3_MAIN configuration port
- Serial flash interface (SFI) features:
  - Serial flash read/write interface
  - Additional registers for defining read and write commands to the external serial flash device
  - 1 to 4 address bytes
  - Fast read support, where fast read requires dummy bytes after address bytes; 0 to 3 dummy bytes can be configured.
  - Dual read support
  - Quad read support
  - Little-endian support only

– Linear increment addressing mode only

The QSPI supports only dual and quad reads. Dual or quad writes are not supported. In addition, there is no "pass through" mode supported where the data present on the QSPI input is sent to its output.

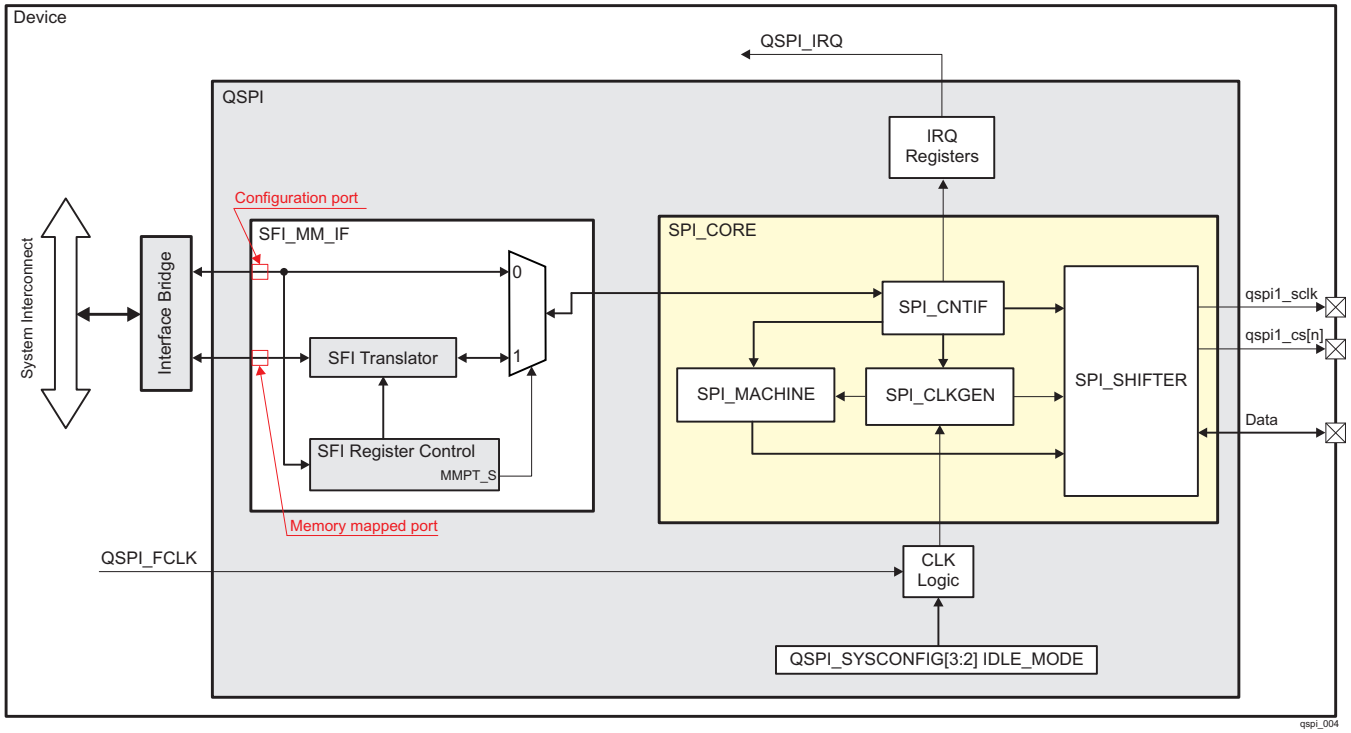
NOTE: The QSPI module does not support cache line wrap mode.

24.2 QSPI Functional Description

24.2.1 QSPI Block Diagram

Iinitial device boot from external SPI flash memory can be accomplished through the QSPI module. The interface is a simple 4-wire SPI used for control or data transfers. The QSPI also supports a 3-wire SPI protocol where the qspi1_d[0] signal is used as a bidirectional for reads and writes. In addition, a 6-wire mode can be used to support quad read devices. Figure 24-2 shows the QSPI block diagram.

Figure 24-2. QSPI Block Diagram



The QSPI is composed of two blocks. The first one is the SFI memory-mapped interface (SFI_MM_IF) and the second one is the SPI core (SPI_CORE). The SFI_MM_IF block is associated only with SPI flash memories and is used for specifying typical for the SPI flash memories settings (read or write command, number of address and dummy bytes, and so on) unlike the SPI_CORE block, which is associated with the SPI interface itself and is used to configure typical SPI settings (chip-select polarity, serial clock inactive state, SPI clock mode, length of the words transferred, and so on).

The SFI_MM_IF comprises the following two subblocks:

- SFI register control
- SFI translator

The SPI_CORE comprises the following four subblocks:

- SPI control interface (SPI_CNTIF)
- SPI clock generator (SPI_CLKGEN)
- SPI control state machine (SPI_MACHINE)

- SPI data shifter (SPI_SHIFTER)

In addition, an interface bridge connects the two ports (configuration port and memory-mapped port) of the SFI_MM_IF block to the L3_MAIN interconnect. There are no software controls associated with this interface bridge.

The QSPI supports long transfers through a frame-style sequence. In its generic SPI use mode, a word can be defined up to 128 bits and multiple words can be transferred during a single access. For each word, a device initiator must read or write the new data and then tell the QSPI to continue the current operation. Using this sequence, a maximum of 4096 128-bit words can be transferred in a single SPI read or write operation. This allows great flexibility when connecting the QSPI to various types of devices.

As opposed to the generic SPI use mode, the communication with serial flash-type devices requires sending a byte command, followed by sending bytes of data. Commands can be sent through the SPI_CORE block to communicate with a serial flash device; however, it is easier to do this using the SFI_MM_IF block because it is intended to ease the communication with serial flash devices. If the SPI_CORE is used to communicate with a serial flash device, software must load the command into the SPI data transfer register with additional configuration fields, perform the byte transfer, then place the data to be sent (or configure for receive) along with additional configuration fields, and perform that transfer. Reads and writes to serial flash devices are more specific. First, the read or write command byte is sent, followed by 1 to 4 bytes of address (corresponding to the address to read/write), then followed by the data write/receive phase. Data is always sent byte oriented. When the address is loaded, data can be continuously read or written, and the address will automatically increment to each byte address internally to the serial flash device.

NOTE: The SFI_MM_IF block only allows reading and writing to an externaly connected SPI flash device. The SFI_MM_IF block does not allow reads or writes to internal configuration and status registers of the SPI flash device. These registers must be accessed through the features of the SPI_CORE block.

24.2.1.1 SFI Register Control

The SFI register control block consists of the following five configuration registers:

- QSPI_SPI_SETUP0_REG
- QSPI_SPI_SETUP1_REG
- QSPI_SPI_SETUP2_REG
- QSPI_SPI_SETUP3_REG
- QSPI_SPI_SWITCH_REG

The first four registers let the user define the following:

- Byte command for a serial flash read specified by the QSPI_SPI_SETUPi_REG[7:0] RCMD bit field
- Byte command for a serial flash write specified by the QSPI_SPI_SETUPi_REG[23:16] WCMD bit field
- Number of address bytes required for the particular type of serial flash specified by the QSPI_SPI_SETUPi_REG[9:8] NUM_A_BYTES bit field
- Number of “dummy bytes” that may be needed to support the fast read mode function of some serial flash devices. The QSPI_SPI_SETUPi_REG[11:10] NUM_D_BYTES bit field specifies the number of "dummy bits." In addition, the QSPI_SPI_SETUPi_REG[28:24] NUM_D_BITS bit field can also specify the number of "dummy bits."
- Whether the read command is single (normal), dual, or quad read mode command. This is specified by the QSPI_SPI_SETUPi_REG[13:12] READ_TYPE bit field. (i is equal to 0, 1, 2 and 3 and means that the QSPI_SPI_SETUPi_REG registers are associated with each of the four supported chip-selects [that is, four supported output SPI flash devices])

The QSPI_SPI_SWITCH_REG register acts as a static switch which allows the configuration port (shown in Figure 24-2) to connect directly to the SPI_CORE block, or allows the memory-mapped port (also shown in Figure 24-2) to connect to the SPI_CORE block. This is done using the QSPI_SPI_SWITCH_REG[0] MMPT_S bit.

In addition, the QSPI_SPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

24.2.1.2 SFI Translator

The SFI translator block represents an FSM which, based on the configuration information loaded into the SFI register control block, converts each input read/write sequence into an SPI_CORE configuration sequence for access to the external serial flash memory.

A read sequence is converted into the following actions:

1. SPI chip-select goes active.
2. Read command byte is issued.
3. 1 to 4 address bytes, which correspond to the first address supplied, are issued.
4. 0 to 3 dummy bytes are issued, if “fast read” is supported.
5. Data bytes are read from the external SPI flash memory.
6. SPI chip-select goes inactive.

For linear addressing mode, action 5 is repeated until the byte count to be transferred reaches zero.

A write sequence is identical to a read sequence, except that a write sequence does not use dummy bytes.

Another important aspect with regard to writes is that a serial flash memory location can only be written to if the bits are erased in advance. Erased means the bits are set to 1. This means that writing only changes 1 contents to 0. It is not possible with this write to change the contents of a bit from 0 to 1. An erase command must be performed to do this operation. Erase commands cannot be executed on single byte locations. Depending on device types, there are page, block, and chip erase commands. To perform an erase command, the particular command must be sent over the SPI bus, and an internal register of the serial flash device must then be polled to determine when the erase completes. The erases must be done through the configuration port by software before performing any writes through the memory-mapped port. This means that writes are passed through to the serial flash device, but if the memory locations being modified are not properly erased before the write, the contents may not result in what was sent.

24.2.1.3 SPI Control Interface

The SPI control interface contains configuration registers used to configure the SPI core functionality of the QSPI. This block maintains all configuration settings for the SPI core (that is, settings specific for the SPI interface itself but not for the SPI flash memories).

The registers defined for this block are:

- The QSPI_PID register, which is read only and contains QSPI revision associated information
- The QSPI_SPI_CLOCK_CNTRL_REG register, which is used to control external SPI clock (qspi1_sclk)
- The QSPI_SPI_DC_REG register used to define the SPI clock mode and chip-select polarity for the four external SPI devices
- The QSPI_SPI_CMD_REG register used to control the operation of the SPI command. This register is also used to configure and transfer data.
- Four data registers used for reading the data received and for writing the data to be transferred. These registers are:
  - QSPI_SPI_DATA_REG
  - QSPI_SPI_DATA_REG_1
  - QSPI_SPI_DATA_REG_2
  - QSPI_SPI_DATA_REG_3

  These four registers compose a 128-bit shift register.
- The QSPI_SPI_STATUS_REG register, which contains status information

All of these registers can only be written if the QSPI is not busy. This means that they can be written if the QSPI_SPI_STATUS_REG[0] BUSY bit is 0x0. The QSPI becomes busy when a write to the QSPI_SPI_CMD_REG[18:16] CMD bit field is performed. Writing to this bit field starts an SPI transaction and sets the QSPI_SPI_STATUS_REG[0] BUSY bit to 0x1. The CMD bit field can be written again when the BUSY bit is 0x0. In addition, the start of the SPI transaction is synchronized to the qspi1_sclk clock and clearing of the BUSY bit is synchronized to the QSPI_FCLK clock.

The register group QSPI_SPI_DATA_REG_3, QSPI_SPI_DATA_REG_2, QSPI_SPI_DATA_REG_1 and QSPI_SPI_DATA_REG is treated as a single 128-bit word for shifting data in and out. The QSPI_SPI_DATA_REG_3 register is used for the most significant bits and the QSPI_SPI_DATA_REG is used for the least significant bits.This applies for both reads and writes. For example, after reading a 128-bit word (WLEN = 0x7F) the most significant bit of the data read, that is bit 127, will be located at QSPI_SPI_DATA_REG_3[31] position and the least significant bit, that is bit 0 of the data read, will be located at the QSPI_SPI_DATA_REG[0] position.

The data written to this register group should be right justified so that a data pre-shifting is not required. The QSPI_SPI_CMD_REG[25:19] WLEN bit field determines the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the WLEN bit field should be set to 0x7 and the data byte should be written to the lower byte of the QSPI_SPI_DATA_REG register. By setting the word length to 0x7 the QSPI_SPI_DATA_REG register will look like a pseudo 8-bit shift register. When the user wants to write 40-bit long word the WLEN bit field should be set to 0x27, the 32 least significant bits of data should be written to the QSPI_SPI_DATA_REG and the rest 8 most significant bits of data should be written to the lower byte of the QSPI_SPI_DATA_REG_1 register. By setting WLEN to 0x27 these two registers will look like a pseudo 40-bit shift register. When the word length is greater than 64 bits the QSPI_SPI_DATA_REG_2 register is also used and the previosly described logic applies. The QSPI_SPI_DATA_REG_3 register is used together with the other three data registers when the word length is greater than 96 bits.

When dual or quad read mode is used the number of the words transferred must be even. This number is configured through the QSPI_SPI_CMD_REG[11:0] FLEN bit field.

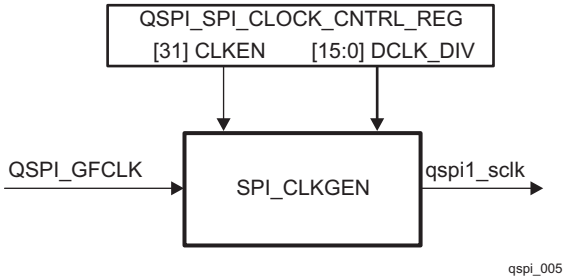
NOTE: The QSPI module does not support a "pass through" mode where the data present on qspi1_d[1] is sent to qspi1_d[0], when 4-pin non-dual read mode is used. This means that setting the QSPI_SPI_CMD_REG[18:16] CMD bit field to 0x1 causes the QSPI only to read from an external device using the qspi1_d[1] pad as an input and if a write to the same external device is desired, the CMD bit field should be set to 0x2, which causes the qspi1_d[0] pad to be used as an output.

24.2.1.4 SPI Clock Generator

The SPI clock generator uses the QSPI_FCLK clock as an input, and generates the qspi1_sclk, which is a divided version of the QSPI_FCLK clock. The divide ratio is a 16-bit value configured through the QSPI_SPI_CLOCK_CNTRL_REG[15:0] DCLK_DIV bit field and thus provides a division factor in a range from 1 to 65536. The QSPI_FCLK clock is divided by the DCLK_DIV value + 1 to provide the qspi1_sclk clock. When DCLK_DIV = 0x0 the QSPI_FCLK clock equals the DCLK clock. The value in the DCLK_DIV bit field applies only when the QSPI_SPI_CLOCK_CNTRL_REG[31] CLKEN bit is set to 0x1. Figure 24-3 shows the SPI_CLKGEN block.

If the CLKEN bit is 0x0 the command specified in the QSPI_SPI_CMD_REG[18:16] CMD bit field is not executed and the QSPI_SPI_STATUS_REG[0] BUSY bit is not set. The command is executed only if the CLKEN bit is 0x1 before write to the CMD bit field.

Figure 24-3. SPI_CLKGEN Block



24.2.1.5 SPI Control State-Machine

The SPI control state-machine (SPI_MACHINE) manages the operation of the SPI_CORE block. SPI_MACHINE takes control and configuration information from the registers in the SPI_CNTIF block as input and provides control information to the SPI data shifter. This information is used to control the SPI data port. The SPI_MACHINE also generates status information, which is sent back to the SPI_CNTIF block.

Writing a valid value to the QSPI_SPI_CMD_REG[18:16] CMD bit field sets immediately the QSPI_SPI_STATUS_REG[0] BUSY bit to 0x1, activates the corresponding qspi1_cs[n] (n = 0 to 3) and starts the SPI data transaction. The BUSY bit is cleared automatically when QSPI_SPI_CMD_REG[25:19] WLEN number of bits are shifted in or out. If the value of the QSPI_SPI_STATUS_REG[27:16] WDCNT bit field is different than 0x0 and WLEN number of bits are shifted already, the SPI_MACHINE waits until another write to the CMD bit field is performed. If the command written to the CMD bit field is valid, then this decrements the value of the WDCNT bit field and starts shfting data in or out again. This is repeated until the WDCNT bit field reaches 0x0, that is, all words of the frame are shifted or till earlier frame termination occurs. While the SPI_MACHINE is waiting for write to the CMD bit field the corresponding qspi1_cs[n] (n = 0 to 3) remains active and the BUSY flag is set to 0x0. In addition, the bit length for each word can be changed during a frame from 1 to 128 bits using the QSPI_SPI_CMD_REG[25:19] WLEN bit field.

The SPI_MACHINE also provides a mechanism to terminate the frame earlier. This is done by writing an invalid command to the CMD bit field. An invalid command corresponds to the 0x0 and 0x4 (reserved) values of the CMD bit field. Writing one of these values when the the WDCNT bit field is not equal to 0x0 and when the BUSY flag is 0x0 terminates the frame earlier.

The corresponding qspi1_cs[n] (n = 0 to 3) becomes inactive when all words are shifted or when the frame terminates earlier.

24.2.1.6 SPI Data Shifter

The SPI data shifter handles the capture and generation of the SPI interface signals. Based on control signals from the SPI_MACHINE and SPI_CNTIF blocks, data is shifted in or out on falling or rising edge of qspi1_sclk clock depending on the SPI clock mode selected. Table 24-1 lists the four defined clock modes of operation for the QSPI.

Table 24-1. SPI Clock Modes Definition

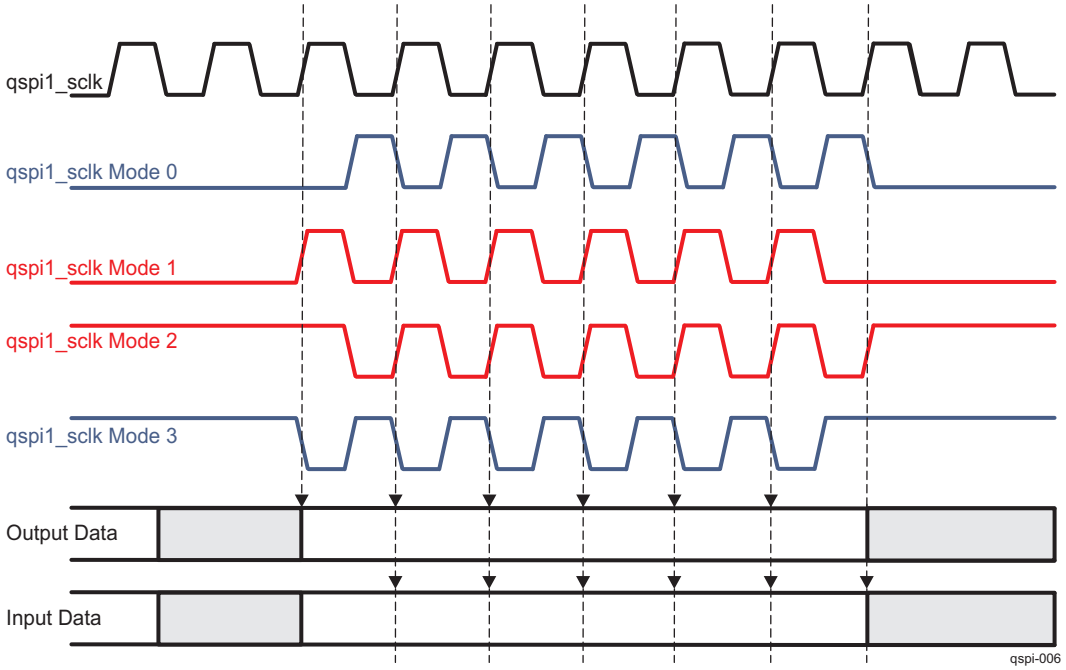
Mode	Settings in the QSPI_SPI_DC_REG Register		Description
	Value of the CKP bits	Value of the CKPH bits	
0	0	0	Data input captured on falling edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock
1	0	1	Data input captured on rising edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock
2	1	0	Data input captured on rising edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock
3	1	1	Data input captured on falling edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock

NOTE: Mode 1 and Mode 2 are not supported and should not be used.

The CKPi and CKPHi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register control the clock modes. Each of these 4 bits corresponds to an output chip select.

Figure 24-4 shows all four clock modes. In addition, through the DDi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register the data can be delayed from one to three qspi1_sclk clock cycles after the corresponding qspi1_cs[n] (n = 0 to 3) goes active. The active state of each chip-select can also be controlled through the CSPi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register.

Figure 24-4. SPI Clock Modes



24.2.2 QSPI Clock Configuration

The QSPI complies with the PRCM slave-idle protocol. The QSPI_FCLK clock is gated based on the values loaded in the QSPI_SYSCONFIG[3:2] IDLE_MODE bit field. Three modes are supported:

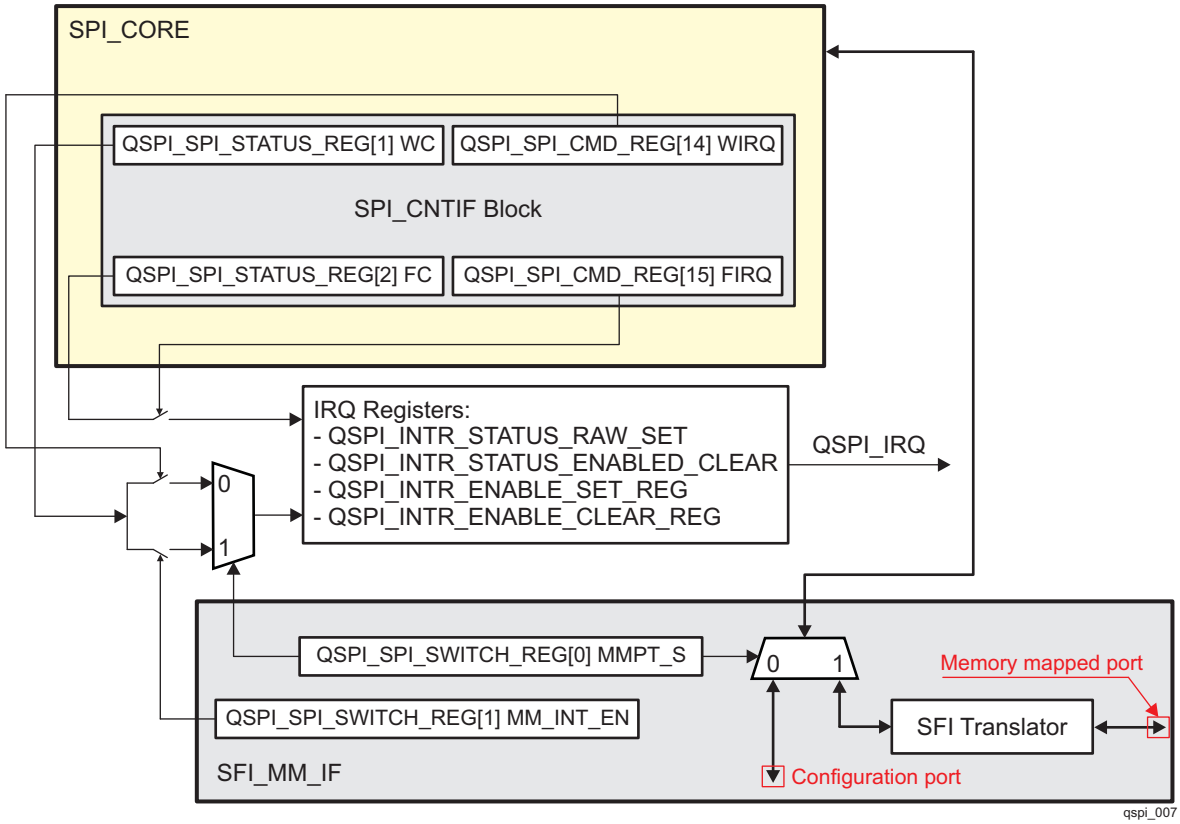
- Force-idle: The QSPI_FCLK clock is gated unconditionally by the QSPI.
- No-idle: The QSPI_FCLK clock is never gated by the QSPI.
- Smart-idle: The QSPI_FCLK clock is gated by the QSPI, depending on its internal requirements.

24.2.3 QSPI Interrupt Requests

The QSPI generates one interrupt request which is connected to the IRQ_CROSSBAR module. This interrupt request, QSPI_IRQ, is connected to the IRQ_CROSSBAR_343 input. The QSPI_IRQ interrupt line can be activated by one of the interrupt events listed in Table 24-2.

Figure 24-5 shows a logical representation of the QSPI interrupt generation scheme.

Figure 24-5. Logical Representation of the QSPI Interrupt Generation Scheme



QSPI_SPI_STATUS_REG[1] WC and QSPI_SPI_STATUS_REG[2] FC are status bits indicating whether word or frame transfer is complete. Setting the corresponding interrupt enable bit (WIRQ or FIRQ) in the QSPI_SPI_CMD_REG register allows these events (WC and FC) to generate an interrupt. The WC and FC bits are reset every time the user writes to the QSPI_SPI_CMD_REG register or reads the QSPI_SPI_STATUS_REG register. This is done to keep control parameters from changing the interface protocol signals while a transfer is in progress. Additionally, the QSPI_SPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

When the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_SPI_CMD_REG[15] FIRQ bits are set to 0x1 the following applies:

- The QSPI activates its interrupt line only if the interrupts are enabled by setting to 0x1 the corresponding bits in the QSPI_INTR_ENABLE_SET_REG register. These interrupts can be disabled by setting the corresponding bits in the QSPI_INTR_ENABLE_CLEAR_REG register to 0x1.
- After an interrupt has been serviced, software must clear the corresponding status flag. This is done by setting the corresponding bit in the QSPI_INTR_STATUS_ENABLED_CLEAR register to 0x1, which also clears the corresponding bit in the QSPI_INTR_STATUS_RAW_SET register. The status flags in the QSPI_INTR_STATUS_RAW_SET register are set even if the corresponding interrupt is disabled unlike those in the QSPI_INTR_STATUS_ENABLED_CLEAR register, which are set only if the corresponding interrupt is enabled.
- The QSPI also generates an interrupt if a certain bit in the QSPI_INTR_STATUS_RAW_SET register is set to 0x1 and the corresponding interrupt is enabled through the QSPI_INTR_ENABLE_SET_REG register. This feature is useful during user software debugging. In addition, even if interrupts are not enabled a corresponding raw flag in the QSPI_INTR_STATUS_RAW_SET register is set to 0x1 when an IRQ condition occurs.
- Even if interrupts are not enabled, a certain status bit in the QSPI_INTR_STATUS_RAW_SET register can also be cleared by setting to 0x1 the corresponding bit in the QSPI_INTR_STATUS_ENABLED_CLEAR register.

It must be considered that the previously described scenario applies if the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_SPI_CMD_REG[15] FIRQ bits are set to 0x1.

NOTE: The QSPI_IRQ interrupt line is activated only if at least one of the following conditions is met:

- The word complete interrupt is enabled:
  - during operations using the memory-mapped port by setting to 0x1 both the QSPI_SPI_SWITCH_REG[1] MM_INT_EN and QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET bits.
  - during operations using the configuration port by setting to 0x1 both the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET bits.
- The frame complete interrupt is enabled setting to 0x1 both the QSPI_SPI_CMD_REG[15] FIRQ and QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET bits.

The QSPI_IRQ interrupt line is also activated when both the conditions are met.

Table 24-2 lists the event flags and the corresponding mask bits of the sources which can cause interrupts.

Table 24-2. QSPI Events

Event Flag	Event Mask	Description
QSPI_INTR_STATUS_RAW_SET[1] WIRQ_RAW	QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET	Word complete interrupt event. Asserted each time after a word is transferred or received.
QSPI_INTR_STATUS_ENABLED_CLEAR[1] WIRQ_ENA	QSPI_INTR_ENABLE_CLEAR_REG[1] WIRQ_ENA_CLR	
QSPI_SPI_STATUS_REG[1] WC	QSPI_SPI_CMD_REG[14] WIRQ	
QSPI_INTR_STATUS_RAW_SET[0] FIRQ_RAW	QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET	Frame complete interrupt event. Asserted each time after a frame is transferred or received.
QSPI_INTR_STATUS_ENABLED_CLEAR[0] FIRQ_ENA	QSPI_INTR_ENABLE_CLEAR_REG[0] FIRQ_ENA_CLR	
QSPI_SPI_STATUS_REG[2] FC	QSPI_SPI_CMD_REG[15] FIRQ	

24.2.4 QSPI Memory Regions

Two memory regions are associated with the QSPI. The first memory region is dedicated to the configuration port. Using this memory region, all internal registers can be programmed and serial transfers made from the external SPI devices. The start address at which the configuration port is available is 0xC0800000. The second memory region is associated mainly with the memory-mapped port and is used for communication directly with external SPI devices. This memory region starts at 0xE0000000 and ends at 0xE0400000.

The CTRL_CORE_CONTROL_IO_2[10:8] QSPI_MEMMAPPED_CS bit field provides a functionality for remapping the previously described address space which starts at 0x5C00 0000 L3_MAIN address to one of the four supported chip selects or to the configuration registers. The CTRL_CORE_CONTROL_IO_2 register resides in the CTRL_MODULE_CORE.

It is important to keep in mind that the configuration port provides an access to all the QSPI registers listed in . These are configuration registers and also four data registers. The configuration registers are used to configure typical SPI and serial flash memory settings and the four data registers are used for read and write operations. When communicating with an external SPI device (but not an SPI flash memory) the SPI_CORE module should be used and the data exchanged is available thorough these four data registers, which can be accessed only through the configuration port. When a communication with an external SPI flash memory is desired, the memory-mapped port should be used.

In other words, to read from an external SPI flash memory, first configure the QSPI through the configuration port and then perform a read through the memory-mapped port.

24.3 QSPI Register Manual

24.3.1 MSS_QSPI Registers

Table 24-3 lists the memory-mapped registers for the MSS_QSPI. All register offset addresses not listed in Table 24-3 should be considered as reserved locations and the register contents should not be modified.

Table 24-3. MSS_QSPI Registers

Offset	Acronym	Register Name	Section
0h	PID	PID	Section 24.3.1.1
10h	SYSCONFIG	SYSCONFIG	Section 24.3.1.2
20h	INTR_STATUS_RAW_SET	INTR Interrupt Status Raw/Set Register	Section 24.3.1.3
24h	INTR_STATUS_ENABLED_CLEAR	INTR Interrupt Status Enabled/Clear Register	Section 24.3.1.4
28h	INTR_ENABLE_SET	INTR Interrupt Enable/Set Register	Section 24.3.1.5
2Ch	INTR_ENABLE_CLEAR	INTR Interrupt Enable/Clear Register	Section 24.3.1.6
30h	INTC_EOI	EOI Register	Section 24.3.1.7
40h	SPI_CLOCK_CNTRL	SPI Clock Control Register (SPICC)	Section 24.3.1.8
44h	SPI_DC	SPI Data Control Register (SPIDC)	Section 24.3.1.9
48h	SPI_CMD	SPI Command Register (SPICR)	Section 24.3.1.10
4Ch	SPI_STATUS	SPI Status Register (SPISR)	Section 24.3.1.11
50h	SPI_DATA	SPI Data Register (SPIDR)	Section 24.3.1.12
54h	SPI_SETUP0	Memory Mapped SPI Setup0 Register	Section 24.3.1.13
58h	SPI_SETUP1	Memory Mapped SPI Setup1 Register	Section 24.3.1.14
5Ch	SPI_SETUP2	Memory Mapped SPI Setup2 Register	Section 24.3.1.15
60h	SPI_SETUP3	Memory Mapped SPI Setup3 Register	Section 24.3.1.16
64h	SPI_SWITCH	Memory Mapped SPI Switch Register	Section 24.3.1.17
68h	SPI_DATA1	SPI Data Register (SPIDR1)	Section 24.3.1.18
6Ch	SPI_DATA2	SPI Data Register (SPIDR2)	Section 24.3.1.19
70h	SPI_DATA3	SPI Data Register (SPIDR3)	Section 24.3.1.20

Complex bit access types are encoded to fit into small table cells. Table 24-4 shows the codes that are used for access types in this section.

Table 24-4. MSS_QSPI Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

24.3.1.1 PID Register (Offset = 0h) [reset = 4F400000h]

PID is shown in Figure 24-6 and described in Table 24-5.

Return to Summary Table.

PID

Figure 24-6. PID Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
R-1h		R-0h		R-F40h			
23	22	21	20	19	18	17	16
FUNC							
R-F40h							
15	14	13	12	11	10	9	8
RTL					MAJOR		
R-0h					R-0h		
7	6	5	4	3	2	1	0
CUSTOM		MINOR					
R-0h		R-0h					

Table 24-5. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	The scheme of the register used. This indicates the PDR3.5 Method
29-28	RESERVED	R	0h	Always read as 0
27-16	FUNC	R	F40h	The function of the module being used
15-11	RTL	R	0h	RTL Release Version The PDR release number of this IP
10-8	MAJOR	R	0h	Major Release Number
7-6	CUSTOM	R	0h	Custom IP
5-0	MINOR	R	0h	Minor Release Number

24.3.1.2 SYSCONFIG Register (Offset = 10h) [reset = 8h]

SYSCONFIG is shown in Figure 24-7 and described in Table 24-6.

Return to Summary Table.

SYSCONFIG

Figure 24-7. SYSCONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED		IDLEMODE		RESERVED	
R-0h		R-0h		R/W-2h		R-0h	

Table 24-6. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Always read as 0
5-4	RESERVED	R	0h	Always read as 0
3-2	IDLEMODE	R/W	2h	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state 0x0 : Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only 0x1 : No-idle mode: local target never enters idle state. Backup mode, for debug only 0x2 : Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events 0x3 : Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented
1-0	RESERVED	R	0h	Always read as 0

24.3.1.3 INTR_STATUS_RAW_SET Register (Offset = 20h) [reset = 0h]

INTR_STATUS_RAW_SET is shown in Figure 24-8 and described in Table 24-7.

Return to Summary Table.

INTR Interrupt Status Raw/Set Register

Figure 24-8. INTR_STATUS_RAW_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						WIRQ_RAW	FIRQ_RAW
R-0h						R/W-0h	R/W-0h

Table 24-7. INTR_STATUS_RAW_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Always read as 0
1	WIRQ_RAW	R/W	0h	Word Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect
0	FIRQ_RAW	R/W	0h	Frame Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect

24.3.1.4 INTR_STATUS_ENABLED_CLEAR Register (Offset = 24h) [reset = 0h]

INTR_STATUS_ENABLED_CLEAR is shown in Figure 24-9 and described in Table 24-8.

Return to Summary Table.

INTR Interrupt Status Enabled/Clear Register

Figure 24-9. INTR_STATUS_ENABLED_CLEAR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						WIRQ_ENA	FIRQ_ENA
R-0h						R/W-0h	R/W-0h

Table 24-8. INTR_STATUS_ENABLED_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Always read as 0
1	WIRQ_ENA	R/W	0h	Word Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect
0	FIRQ_ENA	R/W	0h	Frame Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect

24.3.1.5 INTR_ENABLE_SET Register (Offset = 28h) [reset = 0h]

INTR_ENABLE_SET is shown in Figure 24-10 and described in Table 24-9.

Return to Summary Table.

INTR Interrupt Enable/Set Register

Figure 24-10. INTR_ENABLE_SET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						WIRQ_ENA_SET	FIRQ_ENA_SET
R-0h						R/W-0h	R/W-0h

Table 24-9. INTR_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Always read as 0
1	WIRQ_ENA_SET	R/W	0h	Word Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect
0	FIRQ_ENA_SET	R/W	0h	Frame Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect

24.3.1.6 INTR_ENABLE_CLEAR Register (Offset = 2Ch) [reset = 0h]

INTR_ENABLE_CLEAR is shown in Figure 24-11 and described in Table 24-10.

Return to Summary Table.

INTR Interrupt Enable/Clear Register

Figure 24-11. INTR_ENABLE_CLEAR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						WIRQ_ENA_CLR	FIRQ_ENA_CLR
R-0h						R/W-0h	R/W-0h

Table 24-10. INTR_ENABLE_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Always read as 0
1	WIRQ_ENA_CLR	R/W	0h	Word Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect
0	FIRQ_ENA_CLR	R/W	0h	Frame Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect

24.3.1.7 INTC_EOI Register (Offset = 30h) [reset = 0h]

INTC_EOI is shown in Figure 24-12 and described in Table 24-11.

Return to Summary Table.

EOI Register

Figure 24-12. INTC_EOI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOI_VECTOR																															
R/W-0h																															

Table 24-11. INTC_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EOI_VECTOR	R/W	0h	Number associated with the ipgenericirq for intr output. There are 1 interrupt outputs Write 0x0 : Write to intr IP Generic Any other write value is ignored.

24.3.1.8 SPI_CLOCK_CNTRL Register (Offset = 40h) [reset = 0h]

SPI_CLOCK_CNTRL is shown in Figure 24-13 and described in Table 24-12.

Return to Summary Table.

SPI Clock Control Register (SPICC)

Figure 24-13. SPI_CLOCK_CNTRL Register

31	30	29	28	27	26	25	24
CLKEN	RESERVED						
R/W-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
DCLK_DIV							
R/W-0h							
7	6	5	4	3	2	1	0
DCLK_DIV							
R/W-0h							

Table 24-12. SPI_CLOCK_CNTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLKEN	R/W	0h	Clock Enable. 0- Data clock is turned off 1- Data clock is enabled
30-16	RESERVED	R	0h	Always read as 0
15-0	DCLK_DIV	R/W	0h	Serial data clock divide by ratio

24.3.1.9 SPI_DC Register (Offset = 44h) [reset = 0h]

SPI_DC is shown in Figure 24-14 and described in Table 24-13.

Return to Summary Table.

SPI Data Control Register (SPIDC)

Figure 24-14. SPI_DC Register

31	30	29	28	27	26	25	24
RESERVED			DD3		CKPH3	CSP3	CKP3
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED			DD2		CKPH2	CSP2	CKP2
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED			DD1		CKPH1	CSP1	CKP1
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED			DD0		CKPH0	CSP0	CKP0
R-0h			R/W-0h		R/W-0h	R/W-0h	R/W-0h

Table 24-13. SPI_DC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Always read as 0
28-27	DD3	R/W	0h	Data delay for chip select 3 00- Data is output on the same cycle as the CS_N goes active 01- Data is output 1 DCLK cycle after the CS_N goes active 10- Data is output 2 DCLK cycles after the CS_N goes active 11- Data is output 3 DCLK cycles after the CS_N goes active
26	CKPH3	R/W	0h	Clock phase for chip select 3. If CKP3 = 0: 0- Data shifted out on falling edge; input on rising edge 1- Data shifted out on rising edge; input on falling edge If CKP3 = 1: 1- Data shifted out on falling edge; input on rising edge 0- Data shifted out on rising edge; input on falling edge
25	CSP3	R/W	0h	Chip select polarity for chip select 3 0- Active low 1- Active high
24	CKP3	R/W	0h	Clock polarity for chip select 3 0- When data is not being transferred, SCK = 0 1- When data is not being transferred, SCK = 1
23-21	RESERVED	R	0h	Always read as 0
20-19	DD2	R/W	0h	Data delay for chip select 2 00- Data is output on the same cycle as the CS_N goes active 01- Data is output 1 DCLK cycle after the CS_N goes active 10- Data is output 2 DCLK cycles after the CS_N goes active 11- Data is output 3 DCLK cycles after the CS_N goes active
18	CKPH2	R/W	0h	Clock phase for chip select 2. If CKP2 = 0: 0- Data shifted out on falling edge; input on rising edge 1- Data shifted out on rising edge; input on falling edge If CKP2 = 1: 1- Data shifted out on falling edge; input on rising edge 0- Data shifted out on rising edge; input on falling edge
17	CSP2	R/W	0h	Chip select polarity for chip select 2 0- Active low 1- Active high

Table 24-13. SPI_DC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CKP2	R/W	0h	Clock polarity for chip select 2 0- When data is not being transferred, SCK = 0 1- When data is not being transferred, SCK = 1
15-13	RESERVED	R	0h	Always read as 0
12-11	DD1	R/W	0h	Data delay for chip select 1 00- Data is output on the same cycle as the CS_N goes active 01- Data is output 1 DCLK cycle after the CS_N goes active 10- Data is output 2 DCLK cycles after the CS_N goes active 11- Data is output 3 DCLK cycles after the CS_N goes active
10	CKPH1	R/W	0h	Clock phase for chip select 1. If CKP1 = 0: 0- Data shifted out on falling edge; input on rising edge 1- Data shifted out on rising edge; input on falling edge If CKP1 = 1: 1- Data shifted out on falling edge; input on rising edge 0- Data shifted out on rising edge; input on falling edge
9	CSP1	R/W	0h	Chip select polarity for chip select 1 0- Active low 1- Active high
8	CKP1	R/W	0h	Clock polarity for chip select 1 0- When data is not being transferred, SCK = 0 1- When data is not being transferred, SCK = 1
7-5	RESERVED	R	0h	Always read as 0
4-3	DD0	R/W	0h	Data delay for chip select 0 00- Data is output on the same cycle as the CS_N goes active 01- Data is output 1 DCLK cycle after the CS_N goes active 10- Data is output 2 DCLK cycles after the CS_N goes active 11- Data is output 3 DCLK cycles after the CS_N goes active
2	CKPH0	R/W	0h	Clock phase for chip select 0. If CKP0 = 0: 0- Data shifted out on falling edge; input on rising edge 1- Data shifted out on rising edge; input on falling edge If CKP0 = 1: 1- Data shifted out on falling edge; input on rising edge 0- Data shifted out on rising edge; input on falling edge
1	CSP0	R/W	0h	Chip select polarity for chip select 0 0- Active low 1- Active high
0	CKP0	R/W	0h	Clock polarity for chip select 0 0- When data is not being transferred, SCK = 0 1- When data is not being transferred, SCK = 1

24.3.1.10 SPI_CMD Register (Offset = 48h) [reset = 0h]

SPI_CMD is shown in Figure 24-15 and described in Table 24-14.

Return to Summary Table.

SPI Command Register (SPICR)

Figure 24-15. SPI_CMD Register

31	30	29	28	27	26	25	24
RESERVED		CSNUM		RESERVED		WLEN	
R-0h		R/W-0h		R-0h		R/W-0h	
23	22	21	20	19	18	17	16
WLEN						CMD	
R/W-0h						R/W-0h	
15	14	13	12	11	10	9	8
FIRQ	WIRQ	RESERVED		FLEN			
R/W-0h	R/W-0h	R-0h		R/W-0h			
7	6	5	4	3	2	1	0
FLEN							
R/W-0h							

Table 24-14. SPI_CMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Always read as 0
29-28	CSNUM	R/W	0h	Device select. Sets the active chip select for the transfer 00- Chip Select 0 active 01- Chip Select 1 active 10- Chip Select 2 active 11- Chip Select 3 active
27-26	RESERVED	R	0h	Always read as 0
25-19	WLEN	R/W	0h	Word length. Sets the size of the individual transfers from 1 – 128 bits 0- 1 bit 1- 2 bits … 127 – 128 bits
18-16	CMD	R/W	0h	Transfer command 000- Reserved 001- 4 pin Read Single 010- 4 pin Write Single 011- 4 pin Read Dual 100 – Reserved 101 – 3 pin Read Single 110 – 3 pin Write Single 111 – 6 pin Read Quad
15	FIRQ	R/W	0h	Frame count interrupt enable
14	WIRQ	R/W	0h	Word count interrupt enable
13-12	RESERVED	R	0h	Always read as 0
11-0	FLEN	R/W	0h	Frame Length 0- 1 word 1- 2 words … 4095 – 4096 words

24.3.1.11 SPI_STATUS Register (Offset = 4Ch) [reset = 0h]

SPI_STATUS is shown in Figure 24-16 and described in Table 24-15.

Return to Summary Table.

SPI Status Register (SPISR)

Figure 24-16. SPI_STATUS Register

31	30	29	28	27	26	25	24
RESERVED				WDCNT			
R-0h				R-0h			
23	22	21	20	19	18	17	16
WDCNT							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					FC	WC	BUSY
R-0h					R-0h	R-0h	R-0h

Table 24-15. SPI_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Always read as 0
27-16	WDCNT	R	0h	Word count. This field will reflect the 1-4096 words transferred
15-3	RESERVED	R	0h	Always read as 0
2	FC	R	0h	Frame complete. This bit is set after all of the requested words have been transmitted. 0- Transfer is not complete 1- Transfer is complete This bit is reset when the SPI Status Register is read
1	WC	R	0h	Word complete. This bit is set after each word transfer is completed. 0- Word transfer is not complete 1- Word transfer is complete This bit is reset when the SPI Status Register is read
0	BUSY	R	0h	Busy bit. Active transfer in progress. This bit is only set during an active word transfer. Between words, the bit will clear to signal that it is ok to read/write the data registers. 0- Idle 1- Busy

24.3.1.12 SPI_DATA Register (Offset = 50h) [reset = 0h]

SPI_DATA is shown in Figure 24-17 and described in Table 24-16.

Return to Summary Table.

SPI Data Register (SPIDR)

Figure 24-17. SPI_DATA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R/W-0h																															

Table 24-16. SPI_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Data register for read and write operations

24.3.1.13 SPI_SETUP0 Register (Offset = 54h) [reset = 00020203h]

SPI_SETUP0 is shown in Figure 24-18 and described in Table 24-17.

Return to Summary Table.

Memory Mapped SPI Setup0 Register

Figure 24-18. SPI_SETUP0 Register

31	30	29	28	27	26	25	24
RESERVED			NUM_D_BITS				
R-0h			R/W-0h				
23	22	21	20	19	18	17	16
WCMD							
R/W-2h							
15	14	13	12	11	10	9	8
RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES	
R-0h		R/W-0h		R/W-0h		R/W-2h	
7	6	5	4	3	2	1	0
RCMD							
R/W-3h							

Table 24-17. SPI_SETUP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Always read as 0
28-24	NUM_D_BITS	R/W	0h	Number of dummy bits to use if NUM_D_BYTES = 0
23-16	WCMD	R/W	2h	Write Command
15-14	RESERVED	R	0h	Always read as 0
13-12	READ_TYPE	R/W	0h	Determines if the read command is a single, dual or quad read mode command 00 – Normal read (all data input on spi_din) 01 – Dual read (odd bytes input on spi_din; even on spi_dout) 10 – Normal read (all data input on spi_din) 11 – Quad read (uses spi_qdin0/1)
11-10	NUM_D_BYTES	R/W	0h	Number of dummy bytes to be used for fast read. 0 = use the value in NUM_D_BITS 1 = use 8 bits; 2 = use 16 bits; 3 = use 24 bits
9-8	NUM_A_BYTES	R/W	2h	Number of address bytes to be sent. 0 = 1 byte; 1 = 2 bytes; 2 = 3 bytes; 3 = 4 bytes
7-0	RCMD	R/W	3h	Read Command

24.3.1.14 SPI_SETUP1 Register (Offset = 58h) [reset = 00020203h]

SPI_SETUP1 is shown in Figure 24-19 and described in Table 24-18.

Return to Summary Table.

Memory Mapped SPI Setup1 Register

Figure 24-19. SPI_SETUP1 Register

31	30	29	28	27	26	25	24
RESERVED			NUM_D_BITS				
R-0h			R/W-0h				
23	22	21	20	19	18	17	16
WCMD							
R/W-2h							
15	14	13	12	11	10	9	8
RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES	
R-0h		R/W-0h		R/W-0h		R/W-2h	
7	6	5	4	3	2	1	0
RCMD							
R/W-3h							

Table 24-18. SPI_SETUP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Always read as 0
28-24	NUM_D_BITS	R/W	0h	Number of dummy bits to use if NUM_D_BYTES = 0
23-16	WCMD	R/W	2h	Write Command
15-14	RESERVED	R	0h	Always read as 0
13-12	READ_TYPE	R/W	0h	Determines if the read command is a single, dual or quad read mode command 00 – Normal read (all data input on spi_din) 01 – Dual read (odd bytes input on spi_din; even on spi_dout) 10 – Normal read (all data input on spi_din) 11 – Quad read (uses spi_qdin0/1)
11-10	NUM_D_BYTES	R/W	0h	Number of dummy bytes to be used for fast read. 0 = use the value in NUM_D_BITS 1 = use 8 bits; 2 = use 16 bits; 3 = use 24 bits
9-8	NUM_A_BYTES	R/W	2h	Number of address bytes to be sent. 0 = 1 byte; 1 = 2 bytes; 2 = 3 bytes; 3 = 4 bytes
7-0	RCMD	R/W	3h	Read Command

24.3.1.15 SPI_SETUP2 Register (Offset = 5Ch) [reset = 00020203h]

SPI_SETUP2 is shown in Figure 24-20 and described in Table 24-19.

Return to Summary Table.

Memory Mapped SPI Setup2 Register

Figure 24-20. SPI_SETUP2 Register

31	30	29	28	27	26	25	24
RESERVED			NUM_D_BITS				
R-0h			R/W-0h				
23	22	21	20	19	18	17	16
WCMD							
R/W-2h							
15	14	13	12	11	10	9	8
RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES	
R-0h		R/W-0h		R/W-0h		R/W-2h	
7	6	5	4	3	2	1	0
RCMD							
R/W-3h							

Table 24-19. SPI_SETUP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Always read as 0
28-24	NUM_D_BITS	R/W	0h	Number of dummy bits to use if NUM_D_BYTES = 0
23-16	WCMD	R/W	2h	Write Command
15-14	RESERVED	R	0h	Always read as 0
13-12	READ_TYPE	R/W	0h	Determines if the read command is a single, dual or quad read mode command 00 – Normal read (all data input on spi_din) 01 – Dual read (odd bytes input on spi_din; even on spi_dout) 10 – Normal read (all data input on spi_din) 11 – Quad read (uses spi_qdin0/1)
11-10	NUM_D_BYTES	R/W	0h	Number of dummy bytes to be used for fast read. 0 = use the value in NUM_D_BITS 1 = use 8 bits; 2 = use 16 bits; 3 = use 24 bits
9-8	NUM_A_BYTES	R/W	2h	Number of address bytes to be sent. 0 = 1 byte; 1 = 2 bytes; 2 = 3 bytes; 3 = 4 bytes
7-0	RCMD	R/W	3h	Read Command

24.3.1.16 SPI_SETUP3 Register (Offset = 60h) [reset = 00020203h]

SPI_SETUP3 is shown in Figure 24-21 and described in Table 24-20.

Return to Summary Table.

Memory Mapped SPI Setup3 Register

Figure 24-21. SPI_SETUP3 Register

31	30	29	28	27	26	25	24
RESERVED			NUM_D_BITS				
R-0h			R/W-0h				
23	22	21	20	19	18	17	16
WCMD							
R/W-2h							
15	14	13	12	11	10	9	8
RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES	
R-0h		R/W-0h		R/W-0h		R/W-2h	
7	6	5	4	3	2	1	0
RCMD							
R/W-3h							

Table 24-20. SPI_SETUP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Always read as 0
28-24	NUM_D_BITS	R/W	0h	Number of dummy bits to use if NUM_D_BYTES = 0
23-16	WCMD	R/W	2h	Write Command
15-14	RESERVED	R	0h	Always read as 0
13-12	READ_TYPE	R/W	0h	Determines if the read command is a single, dual or quad read mode command 00 – Normal read (all data input on spi_din) 01 – Dual read (odd bytes input on spi_din; even on spi_dout) 10 – Normal read (all data input on spi_din) 11 – Quad read (uses spi_qdin0/1)
11-10	NUM_D_BYTES	R/W	0h	Number of dummy bytes to be used for fast read. 0 = use the value in NUM_D_BITS 1 = use 8 bits; 2 = use 16 bits; 3 = use 24 bits
9-8	NUM_A_BYTES	R/W	2h	Number of address bytes to be sent. 0 = 1 byte; 1 = 2 bytes; 2 = 3 bytes; 3 = 4 bytes
7-0	RCMD	R/W	3h	Read Command

24.3.1.17 SPI_SWITCH Register (Offset = 64h) [reset = 0h]

SPI_SWITCH is shown in Figure 24-22 and described in Table 24-21.

Return to Summary Table.

Memory Mapped SPI Switch Register

Figure 24-22. SPI_SWITCH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						MM_INT_EN	MMPT_S
R-0h						R/W-0h	R/W-0h

Table 24-21. SPI_SWITCH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Always read as 0
1	MM_INT_EN	R/W	0h	Memory Mapped mode interrupt enable. 0 – Interrupts are disabled during memory mapped operations 1 – Word Count interrupt is enabled for memory mapped operations
0	MMPT_S	R/W	0h	MMPT select. If 0 (default) config port has is selected to control config of core SPI module. If 1, Memory Mapped Protocol Translator is selected to control config port of core SPI module.

24.3.1.18 SPI_DATA1 Register (Offset = 68h) [reset = 0h]

SPI_DATA1 is shown in Figure 24-23 and described in Table 24-22.

Return to Summary Table.

SPI Data Register (SPIDR1)

Figure 24-23. SPI_DATA1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R/W-0h																															

Table 24-22. SPI_DATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Data register for read and write operations

24.3.1.19 SPI_DATA2 Register (Offset = 6Ch) [reset = 0h]

SPI_DATA2 is shown in Figure 24-24 and described in Table 24-23.

Return to Summary Table.

SPI Data Register (SPIDR2)

Figure 24-24. SPI_DATA2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R/W-0h																															

Table 24-23. SPI_DATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Data register for read and write operations

24.3.1.20 SPI_DATA3 Register (Offset = 70h) [reset = 0h]

SPI_DATA3 is shown in Figure 24-25 and described in Table 24-24.

Return to Summary Table.

SPI Data Register (SPIDR3)

Figure 24-25. SPI_DATA3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R/W-0h																															

Table 24-24. SPI_DATA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Data register for read and write operations

Inter-Integrated Circuit (I2C)

This section describes the inter-integrated circuit (I2C or I^2C) module. The I2C is a multi-master communication module providing an interface between the Texas Instruments (TI) microcontroller and devices compliant with Philips Semiconductor I^2C-bus specification version 2.1 and connected by an I2C-bus. This module will support any slave or master I2C compatible device.

Topic **Page**

25.1 Overview **3191**
25.2 I2C Module Operation **3195**
25.3 I2C Operation Modes **3199**
25.4 I2C Module Integrity **3201**
25.5 Operational Information **3203**
25.6 MSS_I2C Registers **3206**
25.7 Sample Waveforms **3233**

25.1 Overview

The I2C has the following features:

- Compliance to the Philips I^{2}C bus specification, v2.1 (*The I2C Specification,* Philips document number 9398 393 40011)
  - Bit/Byte format transfer
  - 7-bit and 10-bit device addressing modes
  - General call
  - START byte
  - Multi-master transmitter/ slave receiver mode
  - Multi-master receiver/ slave transmitter mode
  - Combined master transmit/receive and receive/transmit mode
  - Transfer rates of 10 kbps up to 400 kbps (Phillips fast-mode rate)
- Free data format
- Two DMA events (transmit and receive)
- DMA event enable/disable capability
- Seven interrupts that can be used by the CPU
- Module enable/disable capability
- The SDA and SCL are optionally configurable as general purpose I/O
- Slew rate control of the outputs
- Open drain control of the outputs
- Programmable pullup/pulldown capability on the inputs
- Supports Ignore NACK mode

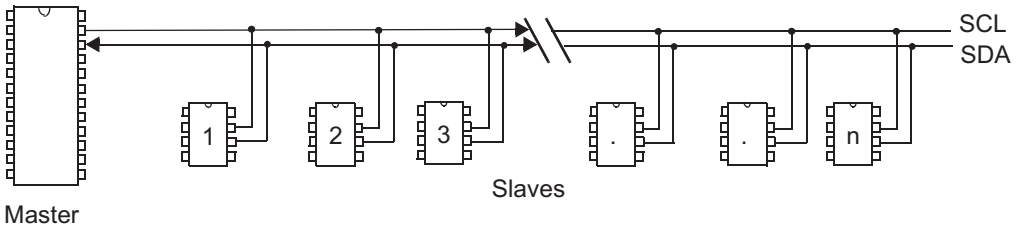
NOTE: This I2C module does **not** support:

- High-speed (HS) mode
- C-bus compatibility mode
- The combined format in 10-bit address mode (the I2C sends the slave address second byte every time it sends the slave address first byte)

25.1.1 Introduction to the I2C Module

The I2C module supports any slave or master I2C-compatible device. Figure 25-1 shows an example of multiple I2C serial ports connected for a two-way transfer from one device to another device.

Figure 25-1. Multiple I2C Modules Connection Diagram



25.1.2 Functional Overview

The I2C module is a serial bus that supports multiple master devices. In multimaster mode, one or more devices can be connected to the same bus and are capable of controlling the bus. Each I2C device on the bus is recognized by a unique address and can operate as either a transmitter or a receiver, depending on the function of the device. In addition to being a transmitter or receiver, a device connected to the I2C bus can also be considered a master or a slave when performing data transfers.

NOTE: A master device is the device that initiates the data transfer on a bus and generates the clock signal that permits the transfer. During the transmission, any device addressed by the master is considered the slave.

Data is communicated to devices interfacing to the I2C module using the serial data pin (SDA) and the serial clock pin (SCL) as shown in Figure 25-2. These two wires carry information between the device and the other devices connected to the I2C bus. Both SDA and SCL pins on the device are bidirectional. They must be connected to a positive supply voltage through a pull-up resistor. When the bus is free, both pins are high. The driver of these two pins has an open-drain configuration to perform the wired-AND function.

The device has a special mode that can be entered to ignore a NACK generated from non-compliant I2C devices that are incapable of generating an ACK.

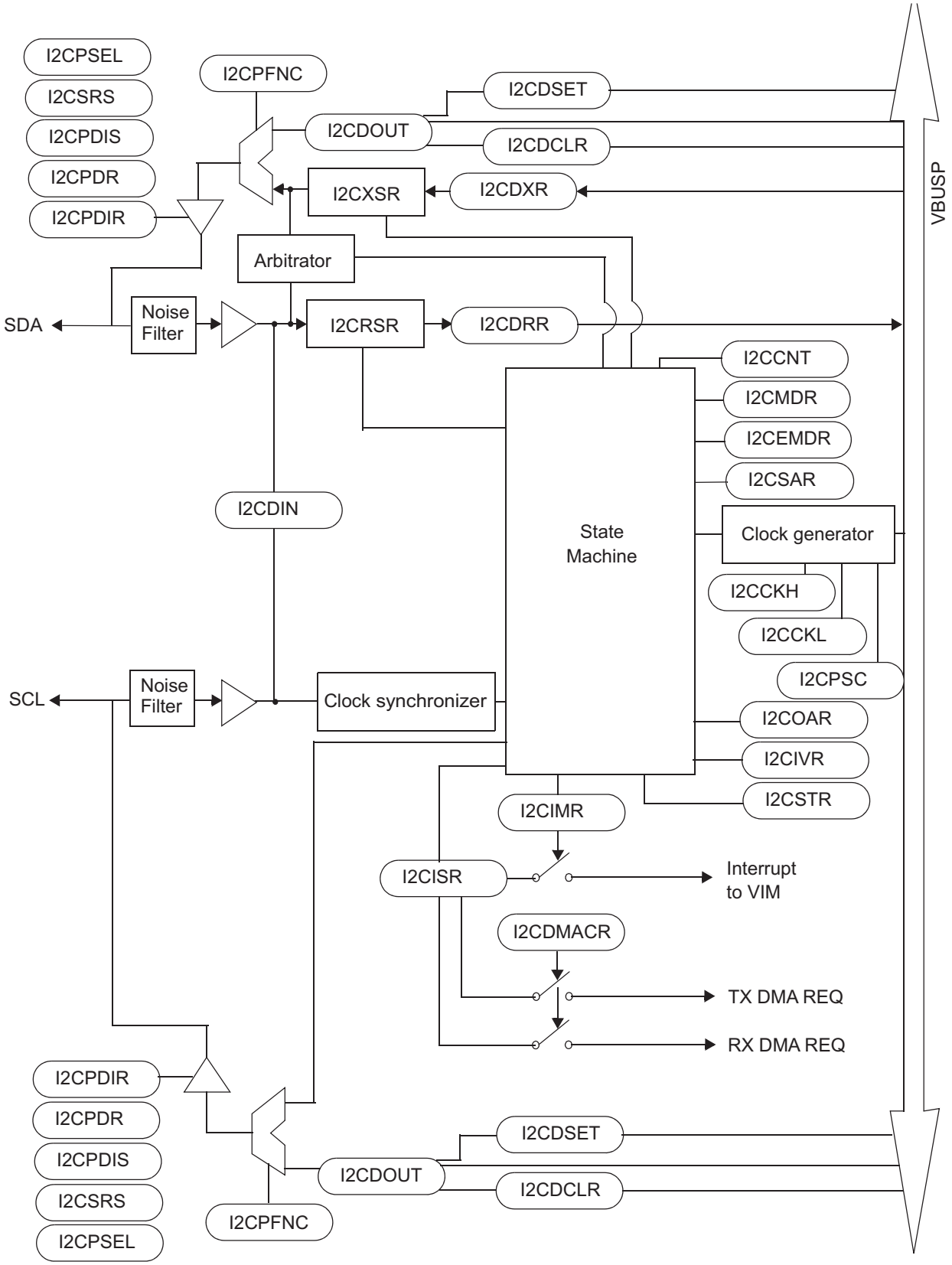
The I2C module consists of the following primary blocks:

- A serial Interface: one data pin (SDA) and one clock pin (SCL)
- The device register interface
  - Data registers to temporarily hold received data and transmitted data traveling between the SDA pin and the CPU or the DMA
  - Control and status registers
- A prescaler to divide down the input clock that is driven to the I2C module
- A peripheral bus interface to enable the CPU and DMA to access the I2C module registers
- An arbitrator to handle arbitration between the I2C module (when configured as a master) and another master
- Interrupt generation logic (interrupts can be sent to the CPU)
- A clock synchronizer that synchronizes the I2C input clock (from the system module) and the clock on the SCL pin, and synchronizes data transfers with masters of different clock speeds.
- A noise filter on each of the two serial pins
- DMA event generation logic that synchronizes data reception and data transmission in the I2C module for DMA transmission

  In Figure 25-2, the CPU or the DMA writes data for transmission to I2CDXR and reads received data from I2CDRR. When the I2C module is configured as a transmitter, data written to I2CDXR is copied to I2CXSR and shifted out one bit at a time. When the I2C module is configured as a receiver, received data is shifted into I2CRSR and then copied to I2CDRR.

  When the I2C function is not needed, the pins may be controlled as general-purpose input/output (GPIO) pins. The I/O structure of each pin includes:
- programmable slew rate control of the outputs
- open drain mode
- programmable pull enable/disable on the input
- programmable pull up/pull down function on the input

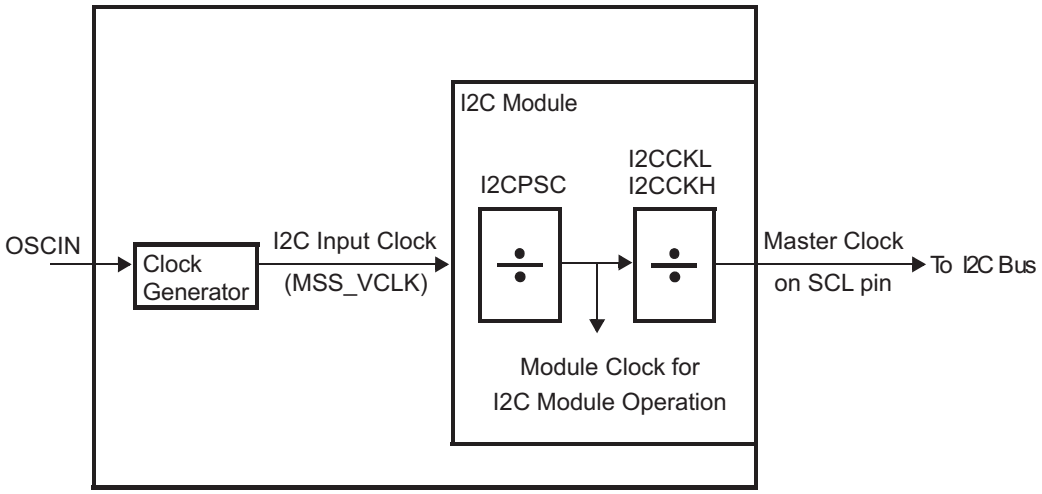
Figure 25-2. Simple I2C Block Diagram



25.1.3 Clock Generation

As shown in Figure 25-3, the I2C module uses the input clock generated from the device clock generator to generate the module clock and master clock. The I2C input clock is the device peripheral clock (VBUS_CLK). The clock is then divided twice more inside the I2C module to produce the module clock and the master clock.

Figure 25-3. Clocking Diagram for the I2C Module



The module clock determines the frequency at which the I2C module operates. A programmable prescaler in the I2C module divides down the input clock to produce the module clock. To specify the divide-down value, initialize the I2CPSC field of the prescaler register, I2CPSC. The resulting frequency is:

$$ModuleClockFrequency = \frac{I2CInputClockFrequency}{(I2CPSC + 1)} \quad (12)$$

The prescaler can only be initialized while the I2C module is in the reset state (IRS = 0 in I2CMDR). The prescaled frequency takes effect only when IRS is changed to 1. Changing the I2CPSC value while IRS = 1 has no effect.

The master clock appears on the SCL pin when the I2C module is configured to be a master on the I2C bus. This clock controls the timing of the communication between the I2C module and a slave. As shown in Figure 25-3, a second clock divider in the I2C module divides down the module clock to produce the master clock. The clock divider uses the I2CCKL to divide down the low portion of the module clock signal and uses the I2CCKH to divide down the high portion of the module clock signal.

The resulting frequency is:

$$MasterClockFrequency = \frac{ModuleClockFrequency}{(I2CCKL + d) + (I2CCKH + d)} \quad (13)$$

$$MasterClockFrequency = \frac{I2CInputClockFrequency}{(I2CPSC + 1)((I2CCKL + d) + (I2CCKH + d))} \quad (14)$$

where *d* depends on the value of I2CPSC:

I2CPSC	d
0	7
1	6
Greater than 1	5

NOTE: The master clock frequency defined above does not include rise/fall time and latency of the synchronizer inside the module. The actual transfer rate will be slower than the value calculated from the formula above. Also, due to the nature of SCL synchronization, the SCL clock period could change if SCL synchronization is taking place.

25.2 I2C Module Operation

The following section discusses how the I2C module operates.

25.2.1 Input and Output Voltage Levels

One clock pulse is generated by the master device for each data bit transferred. Because of a variety of different technology devices that can be connected to the I2C-bus, the levels of logic 0 (low) and logic 1 (high) are not fixed and depend on the associated level of V_{CCIO}. For details, see the device specific data sheet.

25.2.2 I2C Module Reset Conditions

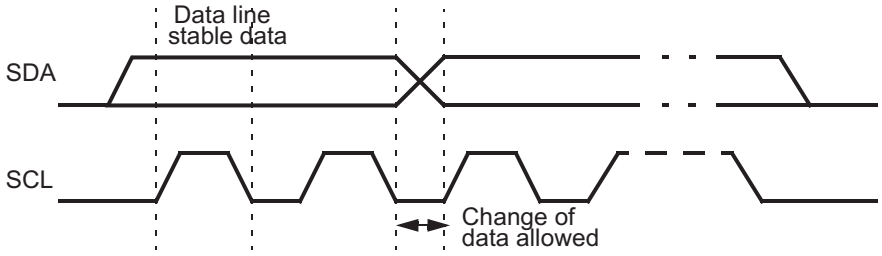
The I2C module can be reset in the following two ways:

- Through the global peripheral reset. A device reset causes a global peripheral reset.
- By clearing the $\overline{\text{IRS}}$ bit in the I2C mode register (I2CMDR). When the global peripheral reset is removed, the $\overline{\text{IRS}}$ bit is cleared to 0, keeping the I2C module in the reset state.

25.2.3 I2C Module Data Validity

The data on the SDA must be stable during the high period of the clock. See Figure 25-4. The high and low state of the data line, the SDA, can only change when the clock signal is low.

Figure 25-4. Bit Transfer on the I2C Bus

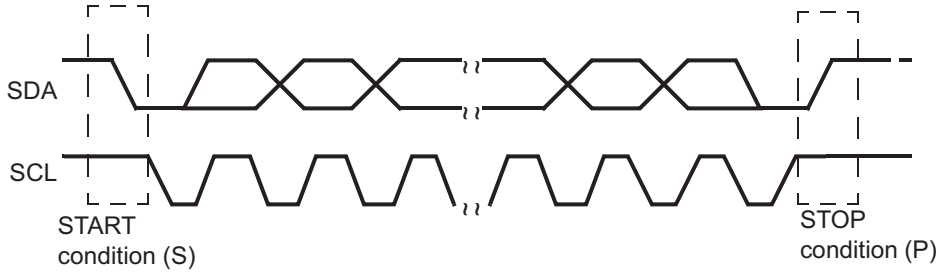


25.2.4 I2C Module Start and Stop Conditions

START and STOP conditions are generated by a master I2C module.

- The START condition is defined as a high-to-low transition on the SDA line while SCL is high. A master drives this condition to indicate the start of data transfer. The bus is considered to be busy after the START condition, and the bus busy bit (BB) in I2CSR is set to 1.
- The STOP condition is defined as a low-to-high transition on the SDA line while SCL is high. A master drives this condition to indicate the end of data transfer. The bus is considered to be free after the STOP condition, therefore the BB bit in I2CSR is cleared to 0.

Figure 25-5. I2C Module START and STOP Conditions

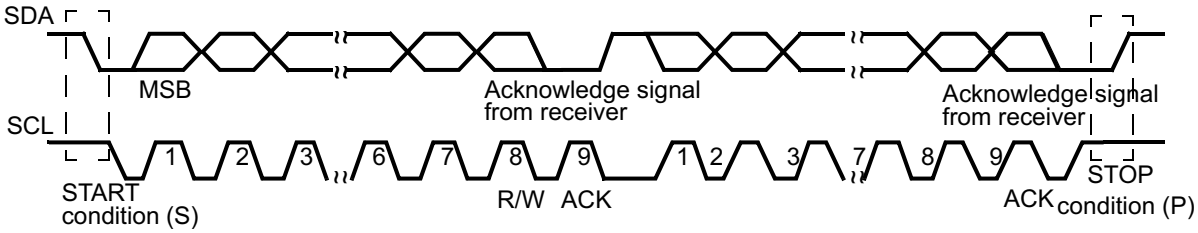


For the I2C module to start a data transfer with a START condition, the master mode bit (MST) and the START condition bit (STT) in the I2CMDR must both be set to 1. For the I2C module to end a data transfer with a STOP condition, the STOP condition bit (STP) must be set to 1. When the BB bit is set to 1 and the STT bit is set to 1, a repeated START condition is generated.

25.2.5 Serial Data Formats

The I2C module operates in byte data format. Each message put on the SDA line is 2 to 8-bits long. The number of messages that can be transmitted or received is unrestricted. The data is transferred with the most significant bit (MSB) first (Figure 25-6). Each message is followed by an acknowledge bit from the I2C if it is in receiver mode. The I2C module does not support little endian systems.

Figure 25-6. I2C Module Data Transfer



The first byte after a START condition (S) always consists of 8 bits that comprise either a 7-bit address plus the R/$\overline{W}$ bit, or 8 data bits. The eighth bit, R/W, in the first byte determines the direction of the data. When the R/$\overline{W}$ bit is 0, the master writes (transmits) data to a selected slave device; when the R/$\overline{W}$ bit is 1, the master reads (receives) data from the slave device. In acknowledge mode, an extra bit dedicated for the acknowledgement (ACK) bit is inserted after each message.

The I2C module supports the following formats:

- 7-bit addressing format (Figure 25-7)
- 10-bit addressing format (Figure 25-8)
- 7-bit/10-bit addressing format with repeated START condition (Figure 25-9)
- Free-data format (Figure 25-10)

25.2.5.1 7-Bit Addressing Format

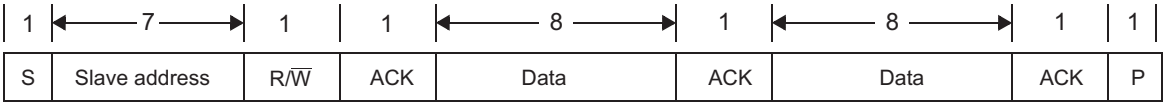
In the 7-bit addressing format (Figure 25-7), the first byte after the START condition consists of a 7-bit slave address followed by the R/$\overline{W}$ bit (in the LSB). The R/$\overline{W}$ bit determines the direction of the data transfer:

- R/$\overline{W}$ = 0: The master writes (transmits) data to the addressed slave.
- R/$\overline{W}$ = 1: The master reads (receives) data from the slave.

An extra clock cycle dedicated for acknowledgement (ACK) is inserted after each byte. If the ACK is inserted by the slave after the first byte from the master, it is followed by n bits of data from the transmitter (master or slave, depending on the R/$\overline{W}$ bit). The device I2C allows n to be a number between 2 to 8, programmable by the bit count (BC) field of I2CMDR. After the data bits have been transferred, the receiver inserts an ACK bit.

To select the 7-bit addressing format, write 0 to the expanded address enable (XA) bit of I2CMDR and make sure the free data format mode is off (FDF = 0 in I2CMDR).

Figure 25-7. I2C Module 7-Bit Addressing Format

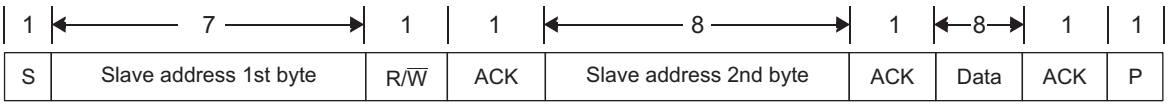


25.2.5.2 10-Bit Addressing Format

The 10-bit addressing format is similar to the 7-bit addressing format, but the master sends the slave address in two separate byte transfers. In the 10-bit addressing format (Figure 25-8), the first byte is 11110b, the two MSBs of the 10-bit slave address, and the R/$\overline{W}$ bit. The ACK bit is inserted after each byte. The second byte is the remaining 8 bits of the 10-bit slave address. The slave must send an acknowledgement after each of the two byte transfers. Once the master has written the second byte to the slave, the master can either write data or use repeated a START condition to change the data direction.

To select the 10-bit addressing format, write 1 to the expanded address enable (XA) bit of I2CMDR and make sure the free data format mode is off (FDF = 0 in I2CMDR).

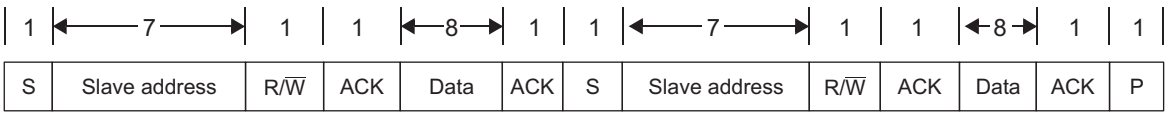
Figure 25-8. I2C Module 10-bit Addressing Format



25.2.5.3 Using the Repeated START Condition

At the end of each byte, the master can drive another START condition (Figure 25-9). Using this capability, a master can transmit/receive any number of data bytes before generating a STOP condition. The length of a data byte can be from 2 to 8 bits. The repeated START condition can be used with the 7-bit addressing, 10-bit addressing, or the free data formats.

Figure 25-9. I2C Module 7-Bit Addressing Format with Repeated START

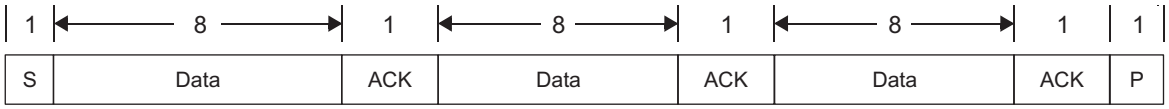


25.2.5.4 Free Data Format

In this format (Figure 25-10), the first byte after a START condition is a data byte. The ACK bit is inserted after each byte, followed by another 8 bits of data. No address or data direction bit is sent. Therefore, the transmitter and receiver must both support the free data format. The direction of data transmission (transmit or receive) remains constant throughout the transfer.

To select the free data format, write a 1 to the free data format (FDF) bit of the I2CMDR. The free data format is not supported in the digital loop back mode.

Figure 25-10. I2C Module in Free Data Format



25.2.6 NACK Bit Generation

When the I2C module is a receiver (master or slave), it can acknowledge or ignore bits sent by the transmitter. To ignore any new bits, the I2C module must send a no-acknowledge (NACK) bit during the acknowledge cycle on the bus. Table 25-1 summarizes the various ways a NACK can be generated.

Table 25-1. Ways to Generate a NACK Bit

I2C Module Condition	Basic NACK Bit Generation Options	Additional Option
Slave receiver mode	Disable data transfers (STT = 0) Allow an overrun condition (RSFULL = 1) Reset the module (IRS = 0)	Set the NACKMOD bit before the rising edge of the last data bit you intend to receive.
Master receiver mode and repeat mode (RM = 1)	Generate a STOP condition (STP = 1) Reset the module (IRS = 0)	Set the NACKMOD bit before the rising edge of the last data bit you intend to receive.
Master receiver mode with non-repeat mode (RM = 0)	If STP = 1, allow the internal data counter to count down to 0 and thus force a STOP condition. If STP = 0, make STP = 1 to generate a STOP condition. Reset the module (IRS = 0)	Set the NACKMOD bit before the rising edge of the last data bit you intend to receive.

In some applications, the slave cannot generate the ACK signal. If the IGNACK bit is set in the I2CEMDR register, the resulting NACK will be ignored and the I2C block will continue the data transfer.

25.3 I2C Operation Modes

25.3.1 Master Transmitter Mode

All masters begin in this mode. The I2C module is a master and transmits control information and data to a slave. In this mode, data assembled in any of the addressing formats shown in Figure 25-7, Figure 25-8, or Figure 25-9 is shifted out onto the SDA pin and synchronized with the self-generated clock pulses on the SCL pin. The clock pulses are inhibited and the SCL pin is held low when the intervention of the device is required ($\overline{XSMT}$ = 0) after a byte has been transmitted.

NOTE: If the I2C is configured for two simultaneous master transmissions, wait until the MST and BB have been reset before performing the second master transmission.

Failure to wait for the MST and BB to reset will prevent the start condition on the second transfer from being issued and the bus BB will not be set. Typically the end of the first transfer is handled by polling BB. However, the MST bit is not reset at the same instant as the BB bit. As a result, when the second master transmission is initiated before the resetting of the MST, the MST bit for the second transfer is reset. This prevents the I2C from recognizing itself as the master, thus failing to occupy the bus.

25.3.2 Master Receiver Mode

In this mode, the I2C module is a master and receives data from a slave. This mode can only be entered from the master transmitter mode (the I2C module must first transmit a command to the slave). In any of the addressing formats shown in Figure 25-7, Figure 25-8, or Figure 25-9, the master receiver mode is entered after the slave address byte and the R/$\overline{W}$ bit have been transmitted (if the R/$\overline{W}$ bit is 1). Serial data bits received on the SDA pin are shifted in with the self-generated clock pulses on the SCL pin. The clock pulses are inhibited and the SCL is held low when the intervention of the device is required (RSFULL = 1) after a byte has been received. At the end of the transfer, the master-receiver signals the end of data to the slave-transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave-transmitter then releases the data line allowing the master-receiver to generate a STOP condition or a repeated START condition.

In many applications, the size of the message is in the initial bytes of the message itself. Since the size of the message is not known to the master before the transmission/reception starts, the master must use the repeat mode in order to force the stop condition when the reception is completed. The repeat mode is enabled by setting the RM bit to 1. Due to the double buffer implementation on the receive side, the master must generate the stop condition (STP =1) after reading the (message size - 1)th data.

25.3.3 Slave Transmitter Mode

In this mode, the I2C module is a slave and transmits data to a master. This mode can only be entered from the slave receiver mode (The I2C module must first receive a command from the master). In any of the addressing formats shown in Figure 25-7, Figure 25-8, or Figure 25-9, the slave transmitter mode is entered if the slave address byte is the same as its own address and the R/$\overline{W}$ bit has been transmitted (if the R/$\overline{W}$ bit is set to 1). The slave transmitter shifts the serial data out on the SDA pin with the clock pulses that are generated by the master device. The slave device does not generate the clock, but it can hold the SCL pin low when intervention of the device is required ($\overline{XSMT}$ = 0) after a byte has been transmitted.

25.3.4 Slave Receiver Mode

In this mode, the I2C module is a slave and receives data from a master. All slaves begin in this mode. Serial data bits received on the SDA pin are shifted in with the clock pulses that are generated by the master device. The slave device does not generate the clock, but it can hold the SCL pin low while intervention of the device is required (RSFULL = 1) after a byte has been received.

25.3.5 Low Power Mode

The I2C module can be placed in low-power mode by a global low-power mode initiated by the system (by writing to the Peripheral Power-Down Set Register in the Peripheral Central Resource (PCR) module.

In effect, low-power mode shuts down all the clocks to the module. In global low-power mode, no registers are visible to the software; nothing can be written to or read from any register.

25.3.6 Free Run Mode

The I2C module can be placed in free run mode when the FREE bit (I2CMDR.14) is set to 1. This bit is primarily used on an emulator when encountering a breakpoint while debugging software. When the FREE bit is set to 0, the I2C responds differently depending on whether the SCL is high or low. If the SCL is low, the I2C stops immediately and keeps driving the SCL low whether the I2C is the master transmitter or receiver. If the SCL is high, the I2C waits until the SCL becomes a low and then stops. If the I2C is a slave, it stops when the transmission/reception completes.

25.3.7 Ignore NACK Mode

The I2C module can be placed in the ignore NACK mode by setting the IGNACK bit in the I2CEMDR register. This mode allows an I2C module that is configured as a master transmitter to ignore a NACK from a slave device that is not capable of generating a proper ACK signal.

25.4 I2C Module Integrity

The following section discusses how the I2C module maintains priorities and order among signals and commands.

25.4.1 Arbitration

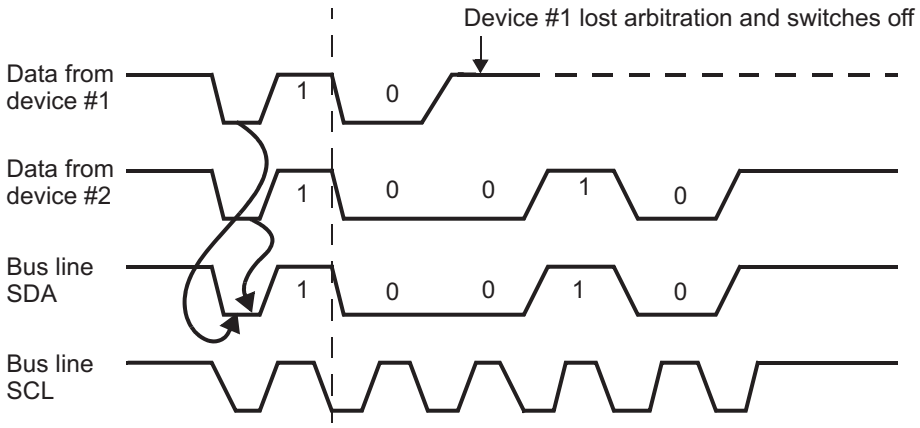
If two or more master transmitters simultaneously start a transmission on the same bus, an arbitration procedure is invoked. Figure 25-11 illustrates the arbitration procedure between two devices. The arbitration procedure uses the data presented on the SDA bus by the competing transmitters. The first master transmitter that generates a high is overruled by the other master that generates a low. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. The master transmitter that loses the arbitration switches to the slave receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration-lost interrupt. The data transmitted by the other master module is salvaged, and the I2C continues to receive data from the master module. Should two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

If, during a serial transfer, the arbitration procedure is still in progress when a repeated START condition or STOP condition is transmitted to I2C bus, the master transmitters involved must send the repeated START condition or STOP condition at the same position in the format frame. In other words, arbitration is not allowed between:

- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

Slaves are not involved in the arbitration procedure.

Figure 25-11. Arbitration Procedure Between Two Master Transmitters



25.4.2 I2C Clock Generation and Synchronization

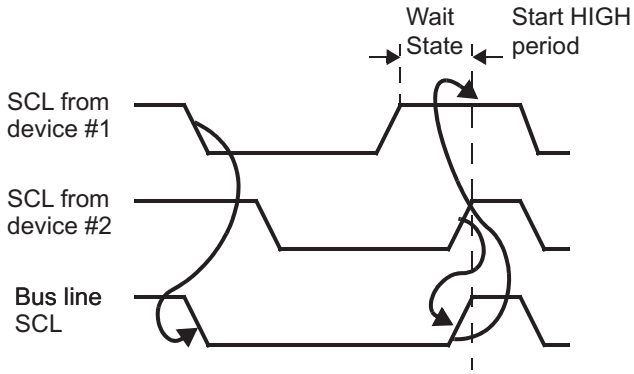
Under normal conditions only one master device generates the clock signal; the SCL. During the arbitration procedure, however, there are two or more master devices and the clock must be synchronized so that the data output can be compared. Figure 25-12 illustrates clock synchronization. The wired-AND property of the SCL line means that a device that first generates a low period on the SCL overrules the other devices. At this high-to-low transition, the clock generators of the other devices are forced to start their own low period. The SCL line is held low by the device with the longest low period. The other devices that finish their low periods must wait for the SCL line to be released before starting their high periods. A synchronized signal on the SCL is obtained where the slowest device determines the length of the low period and the fastest device determines the length of the high period.

If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the wait state. In this way, a slave slows down a fast master and the slow device creates enough time to store a received byte or to prepare a byte to be transmitted.

NOTE: I2C Protocol Fault

The following conditions violate the clock spec as defined in the Philips I²C bus specification, v2.1 (*The I2C Specification,* Philips document number 9398 393 40011), and will result in an I2C protocol fault: I2CCLKH = 2 I2CCLKL = 2I2CPSC = 2. This will cause the SDA data transition to occur while the SCL is high.

Figure 25-12. Synchronization of Two I2C Clock Generators During Arbitration



25.4.3 Prescaler

The I2C module is operated by the module clock. This clock is generated by way of the I2C prescaler block. The prescaler block consists of a 8-bit register, I2CPSC, used for dividing down the device peripheral clock (VBUS_CLK) to obtain a module clock between 6.7 MHz and 13.3 MHz.

25.4.4 Noise Filter

The noise filter is used to suppress any noises that are 50ns or less. It is designed to suppress noise with one module clock, assuming the lower and upper limits of the module clock are 6.7MHz and 13.3MHz, respectively.

25.5 Operational Information

The following section provides specific information about how the I2C module operates.

25.5.1 I2C Module Interrupts

The I2C module generates seven types of interrupts. These seven interrupts are accompanied with seven interrupt mask bits in the interrupt mask register (I2CIMR) and with seven interrupt flag bits in the status register (I2CSR).

25.5.1.1 I2C Interrupt Requests

The I2C module generates the interrupt requests described below. All requests are multiplexed through an arbiter into a single I2C interrupt request to the CPU. Each interrupt request has a flag bit and an enable bit. Interrupts must be enabled prior to the occurrence of the expected interrupt condition. When one of the specified events occurs, the flag bit is set. If the corresponding enable bit is 0, the interrupt request is blocked. If the enable bit is 1, the interrupt request is forwarded to the CPU as an I2C interrupt request. As an alternative, the CPU can poll all of the bits shown in Table 25-2.

Table 25-2. Interrupt Requests Generated by I2C Module

Flag	Name	Generated
AL	Arbitration-lost interrupt	Generated when the I2C module has lost an arbitration contest with another master-transmitter
NACK	No-acknowledge interrupt	Generated when the master I2C does not receive an acknowledge from the receiver
ARDY	Register-access-ready interrupt	Generated when the previously programmed address, data and command have been performed and the status bits have been updated. The interrupt is used to notify the device that the I2C registers are ready to be accessed.
RXRDY	Receive-data-ready interrupt	Generated when the received data in the receive-shift register (I2CSR) has been copied into the data receive register (I2CDRR). The RXRDY bit can also be polled by the device to determine when to read the received data in the I2CDRR.
TXRDY	Transmit-data-ready interrupt	Generated when the transmitted data has been copied from the data transmit register (I2CDXR) into the transmit-shift register (I2CXSR). The TXRDY bit can also be polled by the device to determine when to write the next data into I2CDXR.
SCD	Stop-condition-detect interrupt	Generated when a STOP condition has been detected.
AAS	Address-as-slave interrupt	Generated when the I2C has recognized its own slave address or an address of all zeroes.

The interrupt vector register (I2CIVR) contains the binary-coded-interrupt vector that indicates the highest priority interrupt that is pending and enabled. When I2CIVR is read, the corresponding interrupt flags for AL, NACK and SCD are automatically cleared, if their interrupts are enabled. Reading the I2CIVR will not clear the AAS, ARDY, RXRDY, or TXRDY interrupt pending flags. Please see for the method to clear these four flags.

If more than one interrupt is pending, a new interrupt will be generated for the next highest priority pending interrupt when you re-enable the I2C interrupt.

A transmit interrupt is generated just after the START condition in master transmitter mode. This ensures that the CPU will get an interrupt even if no slave returns an ACK to the slave address following the START condition.

It is important to note that when the I2C is configured to generate interrupts as a slave transmitter and the backward compatibility mode (BCM) bit is set to 1, an extra transmit interrupt occurs. The application should monitor the ACK from the master to determine whether to load another byte into I2CDXR.

25.5.2 DMA Controller Events

The I2C module has two events that use the DMA controller to synchronously read received data (I2CREVNT) from I2CDRR, and synchronously write data (I2CWEVNT) to the transmit buffer, I2CDXR. The read and write events have the same timing as I2CRRDY (I2CRINT) and I2CXRDY (I2CXINT), respectively.

The CPU or the DMA controller reads the received data from I2CDRR and writes the data to be transmitted to I2CDXR. The RXRDY bit is automatically cleared when the DMA controller reads the I2CDRR register, and the TXRDY bit is automatically cleared when the DMA controller writes to the I2CDXR register.

Data written to I2CDXR is copied to I2CXSR and shifted out from the SDA pin when the I2C module is configured as a transmitter. When the I2C module is configured as a receiver, received data is shifted into ICRSR and copied to I2CDRR, which can be read by the CPU or the DMA controller.

A transmit event (I2CWEVNT) is generated after a START condition in master transmitter mode. This ensures that the DMA gets an event even if no slave returns an ACK to the slave address following the START condition.

NOTE: Unexpected DMA transmit and receive event

An unexpected DMA transmit event (ICXEVT) and a DMA receive event (ICXRDY) are generated in 10-bit, master transmit, repeat mode. This event occurs soon after the start condition but before the first bit of the address is transmitted. In this event, no DMA activity should be initiated without the slave ACK being received.

25.5.3 I2C Enable/Disable

The I2C module can be enabled or disabled with the I2C reset enable bit (IRS) in the I2C module register (I2CMDR). This occurs in one of two ways:

- Write 0 to the I2C reset bit (IRS) in I2CMDR. All status bits are forced to the default values and the I2C mode remains disabled until IRS is changed to 1. The SDA and SCL pins are in the high impedance state.
- Initiate a device reset by driving the $\overline{\text{PORRST}}$ pin low. The entire device is reset and is held in the reset state until the pin is released and is driven high. When $\overline{\text{PORRST}}$ is released, all I2C module registers are reset to their default values. The IRS bit is forced to 0, which resets the I2C module. The I2C module stays in the reset state until a 1 is written to the IRS bit.

IRS must be 0 while the I2C module is being configured. Forcing IRS to 0 can be used to save power and also clear error conditions.

25.5.4 General Purpose I/O

Both of the I2C pins can be programmed to be general-purpose I/O pins via the I2C pin control registers (I2CPFNC, I2CDIR, I2CDOUT, and I2CDIN).

When the I2C module is not used, the I2C pins may be programmed to be either general purpose input or general-purpose output pins. This function is controlled in the I2CDIR and I2CPFNC registers. Note that each pin can be programmed to be either an I2C pin or a GIO pin.

If the I2C function is to be used, the application software must ensure that each pin is configured as an I2C pin and not a GIO pin, or else unexpected behavior may result.

25.5.5 Pull Up/Pull Down Function

I2C module pins can have either an active pull up or active pull down that makes it possible to leave the pins unconnected externally. The pins can be programmed to have the active pull function enabled or disabled by writing to the corresponding bit in the I2CPDIS register. Please see the device-specific data sheet for the default internal pull (pull-up, pull-down or no pull) on the pins.

The pull on the pins is programmable to a setting other than the default internal pull as specified in the data sheet. The pins can be programmed to have either an active pull up or an active pull down function by writing to the corresponding bit in I2CPSEL register. The pull up/pull down function is active on the pin only when the pull enabled is programmed in the I2CPDIS register.

The pull up/pull down functions are deactivated when a bidirectional pin is configured as an output. At system reset, the pull up function of all the pins is disabled. Please see the device-specific data sheet for the current supplied by the pull up/pull down.

25.5.6 Open Drain Function

The I2C pins can be programmed to include an open drain function when they are configured as output pins. This is done by writing to the corresponding bit of the I2CPDR register. When the open drain function is enabled, a low value (0) written to the data output register forces the pin to a low output voltage (V_{OL} or lower), whereas a high value (1) written to the data output register forces the pin to a high-impedance state. The open drain function is disabled when the pin is configured as an input pin.

25.6 MSS_I2C Registers

Table 25-3 lists the memory-mapped I2C registers. All register offset addresses not listed in Table 25-3 should be considered as reserved locations and the register contents should not be modified. The upper word (upper 16 bits) of the registers all read as 0s. Writes have no effect on these bits. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

Table 25-3. MSS_I2C Registers

Offset	Acronym	Register Name	Section
0h	ICOAR	ICOAR	Section 25.6.1
4h	ICIMR	ICIMR	Section 25.6.2
8h	ICSTR	ICSTR	Section 25.6.3
Ch	ICCLKL	ICCLKL	Section 25.6.4
10h	ICCLKH	ICCLKH	Section 25.6.5
14h	ICCNT	ICCNT	Section 25.6.6
18h	ICDRR	ICDRR	Section 25.6.7
1Ch	ICSAR	ICSAR	Section 25.6.8
20h	ICDXR	ICDXR	Section 25.6.9
24h	ICMDR	ICMDR	Section 25.6.10
28h	ICIVR	ICIVR	Section 25.6.11
2Ch	ICEMDR	ICEMDR	Section 25.6.12
30h	ICPSC	ICPSC	Section 25.6.13
34h	ICPID1	ICPID1	Section 25.6.14
38h	ICPID2	ICPID2	Section 25.6.15
3Ch	ICDMAC	ICDMAC	Section 25.6.16
48h	ICPFUNC	ICPFUNC	Section 25.6.17
4Ch	ICPDIR	ICPDIR	Section 25.6.18
50h	ICPDIN	ICPDIN	Section 25.6.19
54h	ICPDOUT	ICPDOUT	Section 25.6.20
58h	ICPDSET	ICPDSET	Section 25.6.21
5Ch	ICPDCLR	ICPDCLR	Section 25.6.22
60h	ICPDRV	ICPDRV	Section 25.6.23

Complex bit access types are encoded to fit into small table cells. Table 25-4 shows the codes that are used for access types in this section.

Table 25-4. MSS_I2C Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

25.6.1 ICOAR Register (Offset = 0h) [reset = 0h]

ICOAR is shown in Figure 25-13 and described in Table 25-5.

Return to Summary Table.

I2C Own Address register

Figure 25-13. ICOAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																						A9_A0									
R/W-0h																						R/W-0h									

Table 25-5. ICOAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU	R/W	0h	Reserved
9-0	A9_A0	R/W	0h	Own address. Use in both 7- and 10-bit address mode. Note that usercan program the I2C own address to any value as long as it does notconflict with other components in the system.

25.6.2 ICIMR Register (Offset = 4h) [reset = 0h]

ICIMR is shown in Figure 25-14 and described in Table 25-6.

Return to Summary Table.

I2C Interrupt Mask/Status register

Figure 25-14. ICIMR Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU	AAS	SCD	ICXRDY	ICRRDY	ARDY	NACK	AL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 25-6. ICIMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	NU	R/W	0h	Reserved
6	AAS	R/W	0h	Address As Slave interrupt mask bit. Setting a"1" to this bit unmasks the Address As Slave interrupt. Setting a"0" to this bit masks the Address As Slave interrupt.
5	SCD	R/W	0h	Stop Condition Detection mask bit. Setting a"1" to this bit unmasks the Stop Condition Detection interrupt. Setting a "0" to this bit masks the Stop Condition Detection interrupt.
4	ICXRDY	R/W	0h	Transmit Data Ready interrupt mask bit. Setting a"1" to this bit unmasks the Transmit Data Ready interrupt. Setting a"0" to this bit masks the Transmit Data Ready interrupt.
3	ICRRDY	R/W	0h	Receive Data Ready interrupt mask bit. Setting a"1" to this bit unmasks the Receive Data Ready interrupt. Setting a"0" to this bit masks the Receive Data Ready interrupt.
2	ARDY	R/W	0h	Register access ready interrupt mask bit. Setting a"1" to this bit unmasks the Register access ready interrupt. Setting a"0" to this bit masks the Register access ready interrupt.
1	NACK	R/W	0h	No Acknowledgement interrupt mask bit. Setting a"1" to this bit unmasks the No Acknowledgement interrupt. Setting a"0" to this bit masks the No Acknowledgement interrupt.
0	AL	R/W	0h	Arbitration Lost interrupt mask bit. Setting a"1" to this bit unmasks the Arbitration Lost interrupt. Setting a"0" to this bit masks the Arbitration Lost interrupt.

25.6.3 ICSTR Register (Offset = 8h) [reset = 0h]

ICSTR is shown in Figure 25-15 and described in Table 25-7.

Return to Summary Table.

I2C Interrupt Status register

Figure 25-15. ICSTR Register

31	30	29	28	27	26	25	24
NU2							
R/W-0h							
23	22	21	20	19	18	17	16
NU2							
R/W-0h							
15	14	13	12	11	10	9	8
NU2	SDIR	NACKSNT	BB	RSFULL	XSMT	AAS	AD0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
NU1		SCD	ICXRDY	ICRRDY	ARDY	NACK	AL
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 25-7. ICSTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	NU2	R/W	0h	Reserved
14	SDIR	R/W	0h	Slave Direction. This bit is clear to '0' indicating the I2C is a master transmitter/receiver or a slave receiver. This bit is also clear by STOP condition or START condition. It is set to '1' when the I2C slave is a transmitter. In DLB mode (which the configuration should be master-transmitter slave-receiver) this bit is clear to '0'. Writing a"1" to this bit to clear it.
13	NACKSNT	R/W	0h	A No Acknowledge is sent due to NACKMOD is set to a"1". NACKSNT =0: A No Acknowledge is not sent. NACKSNT =1: A No Acknowledge is sent. Writing a"1" to this bit to clear it.
12	BB	R/W	0h	Bus Busy. This bit indicates the state of the serial bus. BB=0: The bus is free. BB=1: The bus is occupied. On reception of a"start" condition the device sets BB to 1. This bit is also set if the I2C detects SCL low state. BB is clear to 0 after reception of a"stop" condition. BB is kept to"0" regardless SCL state when the I2C is in reset (IRS_=0). If the IRS_ is set to"1" during transaction between other I2C devices the BB bit is set at the first falling edge of SCL or START condition. - (RW)
11	RSFULL	R/W	0h	Receive shift full. This bit indicates whether the receiver has experienced overrun. Overrun occurs when the receive shift register (ICRSR) is full and ICDRR has not been read since the ICRSR-to-ICDRR transfer. The FSM is holding for ICDRR read access. RSFULL is clear when reading the ICDRR. RSFULL is set to"1" when the I2C has recognized an overrun. The contents of ICDRR are NOT lost in this case. In repeat mode since double buffer (ICRSR and ICDRR) behaves like a single buffer RSFULL is set to"1" every time the data is received. RSFULL is clear as a result of reading the ICDRR. - (RW)
10	XSMT	R/W	0h	Transmit shift empty not. This bit indicates whether the transmitter has experienced underflow. Underflow occurs when the transmit shift register (ICXSR) is empty and ICDXR has not been loaded. The FSM is holding for ICDXR write access. XSMT_ is cleared when underflow has occurred. XSMT_ is set to"1" as a result of writing to ICDXR. In repeat mode if the I2C in master transmitter mode is holding transfer with XSMT_=0 (i.e. waiting for further action) and the STT or STP bit is set XSMT_ is set to"1" by hardware.

Table 25-7. ICSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	AAS	R/W	0h	Address As Slave. This bit is set to 1 by the device when it has recognized its own slave address or an address of all (8) zeros. The AAS bit is reset by stop condition or detection of any address byte that does not match ICOAR. - (RW)
8	AD0	R/W	0h	Address Zero Status: This bit is set to 1 by device if it detects the address of all (8) zeros (i.e. general call). The AD0 bit is reset to 0 (default value) when a"start" or"stop" condition is detected. - (RW)
7-6	NU1	R/W	0h	Reserved
5	SCD	R/W	0h	Stop Condition Detection bit SCD is set when the I2C sends or receives STOP condition. This bit is cleared by reading ICIVR (as 110) or writing '1' to itself.
4	ICXRDY	R/W	0h	Transmit Data Ready interrupt flag bit. ICXRDY is set to"1" is generated when the transmitted data has been copied from ICDXR to the transmit-shift register (ICXSR). ICRXDY is clear to"0" when the ICDXR is written. This bit can also be polled by the CPU to write a new transmitted data into the ICDXR. Write '1' to this bit will set it and DXR Write will clear it.
3	ICRRDY	R/W	0h	Receive Data Ready interrupt flag bit. ICRRDY is set to"1" when the received data has been copied from ICRSR into the ICDRR. ICRRDY is cleared to"0" when the ICDRR is read. This bit can also be polled by the CPU to read the received data in the ICDRR. Write '1' or DRR Read will clear it.
2	ARDY	R/W	0h	Register-access-ready interrupt flag bit. ARDY is generated by the hardware if the I2C is in the master mode when the previously programmed data and command has been performed and status bit has been updated. This flag is used by the CPU to let it knows that the I2C registers are ready to be accessed again. When RM=0 ARDY is set when the internal data count is passed 0 if STP register bit has not been set. When RM=1 ARDY is set at each byte end. If the I2C is in FDF mode(FDF=1) ARDY is set just after Start condition. This bit is automatically cleared by hardware when writing data to ICDXR in transmit mode reading data from ICDRR in receive mode or setting STT or STP bit. Write '1' will clear it.
1	NACK	R/W	0h	No-Acknowledgement interrupt flag bit. The No Acknowledge flag bit is set when the hardware in "master" mode detects no acknowledge has been received. This bit is NOT set by no-acknowledgement after Start byte Write '1' or Read the ICIVR (as 010) will clear it.
0	AL	R/W	0h	Arbitration-Lost interrupt flag bit. The Arbitration Lost flag bit is set to 1 when the device in the "master" mode senses it has lost an arbitration when two or more transmitters start a transmission almost simultaneously or when the I2C attempts to start a transfer while BB (bus busy) is 1. When this is set to 1 due to arbitration lost the MST/STT/STP bits are clear the I2C becomes a slave. Write '1' or Read the ICIVR (as 001) will clear it.

25.6.4 ICCLKL Register (Offset = Ch) [reset = 0h]

ICCLKL is shown in Figure 25-16 and described in Table 25-8.

Return to Summary Table.

I2C Clock Divider Low register

Figure 25-16. ICCLKL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																ICCL15_ICCL0															
R/W-0h																R/W-0h															

Table 25-8. ICCLKL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved
15-0	ICCL15_ICCL0	R/W	0h	Low time I 2 C SCL Clock Division Factor. They are used to divide down the master clock to create the SCL low time transition frequency. This register must be configured while the I2C is still in reset (IRS_=0).

25.6.5 ICCLKH Register (Offset = 10h) [reset = 0h]

ICCLKH is shown in Figure 25-17 and described in Table 25-9.

Return to Summary Table.

I2C Clock Divider High register

Figure 25-17. ICCLKH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																ICCH15_ICCLH0															
R/W-0h																R/W-0h															

Table 25-9. ICCLKH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved
15-0	ICCH15_ICCLH0	R/W	0h	High time I 2 C SCL Clock Division Factor. They are used to divide down the master clock to create the SCL high time transition frequency. This register must be configured while the I2C is still in reset (IRS_=0).

25.6.6 ICCNT Register (Offset = 14h) [reset = 0h]

ICCNT is shown in Figure 25-18 and described in Table 25-10.

Return to Summary Table.

I2C Data Count register

Figure 25-18. ICCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																ICDC15_ICDC0															
R/W-0h																R/W-0h															

Table 25-10. ICCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved
15-0	ICDC15_ICDC0	R/W	0h	Data count. This data count register is used to generate a Stop condition if a Stop condition is specified (STP=1). . ICCNT=1 data count is 1 :: :: ICCNT=0FFFFh data count is 65535 ICCNT=0data counter is 65536 Note that ICCNT is a don"t care when RM is set to 1.

25.6.7 ICDRR Register (Offset = 18h) [reset = 0h]

ICDRR is shown in Figure 25-19 and described in Table 25-11.

Return to Summary Table.

I2C Data Receive register

Figure 25-19. ICDRR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								D7_D0							
R/W-0h																								R/W-0h							

Table 25-11. ICDRR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved
7-0	D7_D0	R/W	0h	Receive data

25.6.8 ICSAR Register (Offset = 1Ch) [reset = 0h]

ICSAR is shown in Figure 25-20 and described in Table 25-12.

Return to Summary Table.

I2C Slave Address register

Figure 25-20. ICSAR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																						A9_A0									
R/W-0h																						R/W-0h									

Table 25-12. ICSAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU	R/W	0h	Reserved
9-0	A9_A0	R/W	0h	Slave address. Use in both 7- and 10-bit address mode.

25.6.9 ICDXR Register (Offset = 20h) [reset = 0h]

ICDXR is shown in Figure 25-21 and described in Table 25-13.

Return to Summary Table.

I2C Data Transmit register

Figure 25-21. ICDXR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								D7_D0							
R/W-0h																								R/W-0h							

Table 25-13. ICDXR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved
7-0	D7_D0	R/W	0h	Transmit data

25.6.10 ICMDR Register (Offset = 24h) [reset = 0h]

ICMDR is shown in Figure 25-22 and described in Table 25-14.

Return to Summary Table.

I2C Mode register

Figure 25-22. ICMDR Register

31	30	29	28	27	26	25	24
NU2							
R/W-0h							
23	22	21	20	19	18	17	16
NU2							
R/W-0h							
15	14	13	12	11	10	9	8
NACKMOD	FREE	STT	NU1	STP	MST	TRX	XA
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RM	DLB	IRS	STB	FDF	BC2_BC1_BC0		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

Table 25-14. ICMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU2	R/W	0h	Reserved
15	NACKMOD	R/W	0h	No Acknowledge (NACK) mode. This bit is used to send an Acknowledge (ACK) or a No Acknowledge (NACK) to the transmitter. This bit is only applicable when the I2C is in receiver mode. In master receiver mode when the internal data count counter decrements to zero the I2C sends a NACK. The master receiver I2C finishes a transfer when it sends a NACK. The I2C ignores ICCNT when NACKMOD is '1'. The NACKMOD bit should be set before the rising edge of the last data bit (bit 8) if a NACK must be sent and this bit is cleared once a NACK has been sent. NACKMOD=0 the I2C sends an ACK to the transmitter during the acknowledge cycle. NACKMOD=1 the I2C sends a NACK to the transmitter during the acknowledge cycle.
14	FREE	R/W	0h	Free Running. This bit is used to determine the state of the I2C when a breakpoint is encountered in the HLL debugger. FREE=0: (default) Stops immediately if SCL is low and keep driving SCL low whether I2C is master transmitter/receiver. If SCL is high I2C waits until SCL becomes low and then stops. If the I2C is a slave it will stop when the transmission/receiving completes. FREE=1: The I2C runs free.
13	STT	R/W	0h	Start Condition (Master only mode). This bit can be set to a"1" by the CPU to generate a Start condition. In master mode when setting Start to"1" generates a Start condition. It is reset to "0" by the hardware after the Start condition has been generated. The Start/Stop bits can be configured to generate different transfer formats. Note that the STT and STP can be used to terminate the repeat mode. STT___STP____Conditions_____Bus Activities _1_____0_____Start_____S-A-D _0_____1_____Stop_____P _1_____1_____Start-Stop (ICCNT= n)_____S-A-D..(n)..D-P _1_____0_____Start (ICCNT= n)_____S-A-D..(n)..D
12	NU1	R/W	0h	Reserved for IDLEEN (IDLE Enable on 5509). - (RW)

Table 25-14. ICMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	STP	R/W	0h	Stop Condition (Master mode only). This bit can be set to a"1" by the CPU to generate a Stop condition. It is reset to "0" by the hardware after the Stop condition has been generated. The Stop condition is generated when ICCNT passes 0 when the I2C is in non-repeat mode(RM=0).
10	MST	R/W	0h	Master. MST=0: The I 2 C peripheral is in the"slave" mode and clock is received from the"master" device. MST=1: The I 2 C peripheral is in the"master" mode and it generates the clock. This bit is clear when the transfer completed.
9	TRX	R/W	0h	Transmitter. TRX=0: The I 2 C is in the"receiver" mode and data on data line SDA is shifted into the data register ICDRR. TRX=1: The I 2 C is in the"transmitter" mode and the data in ICDXR is shifted out on data line SDA. The operating modes (not in FDF mode) are defined as follows. In FDF mode TRX must be configured even if the I2C is in slave mode because there is no address/direction byte in FDF mode. _____ MST___TRX___Operating Modes _0_____x_____"slave receiver" _0_____x_____"slave transmitter" _1_____0_____"master receiver" _1_____1_____"master transmitter" _____
8	XA	R/W	0h	Expanded Address. XA=0: (default) 7-bit address mode (normal address mode). XA=1: 10-bit address mode (expanded address mode) Please note that XA needs to be configured even if the I2C is in slave mode.
7	RM	R/W	0h	Repeat Mode. This bit is set to a"1" by the CPU to put the I2C in the repeat mode. In this mode data is continuously transmitted out of the ICDXR until the STP bit is set to"1" regardless of ICCNT value. This bit is don"t care if the I2C is configured in slave mode. _____ __ RM___STT___STP___Conditions_____Bus Activities_____Mode _0_____0_____0_____Idle_____None_____NA _0_____0_____1_____Stop_____P_____NA _0_____1_____0_____(Re)Start_____S-A-D..(n)..D____Repeat n _0_____1_____1_____(Re)Start-Stop___S-A-D..(n)..D-P__Repeat n _1_____0_____0_____Idle_____none_____NA _1_____0_____1_____Stop_____P _____NA _1_____1_____0_____(Re)Start_____S-A-D-D-D..._____Continuous _1_____1_____1_____Reserved_____None_____NA _____ __
6	DLB	R/W	0h	Digital Loop Back (in master transmit mode only). This bit is set to a"1" by the CPU to put the I2C in the loop back mode. In this mode data transmitted out of the ICDXR will be received in the ICDRR after ((CPU freq/I2C freq)8) CPU cycles via an internal path. The address of the ICOAR is output on SDA.
5	IRS	R/W	0h	I2C Reset Not. This can be set to a"0" by the CPU to put the I2C in reset or to a"1" to take the I2C out of reset. When this bit is reset to 0 all status bits in ICSTR and ICIVR are set to default values. Note that if this bit is reset during a transfer it can cause the I2C bus hang (SDA and SCL are tri-stated).
4	STB	R/W	0h	Start Byte (Master only mode). The Start Byte mode bit is set to 1 by the CPU to configure the I2C in Start byte mode the I2C sends "00000001" regardless ICSAR value. Refer to the Philip I2C spec for more details.

Table 25-14. ICMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	FDF	R/W	0h	Free Data Format. This bit can be set to"1" by the CPU to configure the I2C in Free Data Format mode. FDF___MST___TRX_____Operating mode _0_____0_____ x____Slave in non FDF mode _0_____1_____0____Master receive in non FDF mode _0_____1_____1____Master transmit in non FDF mode _1_____0_____0____Slave receiver in FDF mode _1_____0_____1____Slave transmitter in FDF mode _1_____1_____0____Master receiver in FDF mode _1_____1_____1____Master transmitter in FDF mode
2-0	BC2_BC1_BC0	R/W	0h	Bit Count : Bit Count 2, Bit Count 1 and Bit Count 0 define the number of bits starting from the lsb (excluding the acknowledge bit) of the next byte which are yet to be received or transmitted. BC2_BC1_BC0__Bits/byte in FDF__Bits/byte w/ ACK _0___0___1_____NA (reserved)____ NA (reserved) _0___1___0_____2_____3_____ _0___1___1_____3_____4_____ _1___0___0_____4_____5_____ _1___0___1_____5_____6_____ _1___1___0_____6_____7_____ _1___1___1_____7_____8_____ _0___0___0_____8_____9_____

25.6.11 ICIVR Register (Offset = 28h) [reset = 0h]

ICIVR is shown in Figure 25-23 and described in Table 25-15.

Return to Summary Table.

I2C Interrupt Vector register

Figure 25-23. ICIVR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU2															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2				TESTMD				NU1					INTCODE		
R/W-0h				R/W-0h				R/W-0h					R/W-0h		

Table 25-15. ICIVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU2	R/W	0h	Reserved.
11-8	TESTMD	R/W	0h	Reserved for internal testing.
7-3	NU1	R/W	0h	Reserved.
2-0	INTCODE	R/W	0h	Interrupt code. The binary-coded-interrupt vector indicates which interrupt has occurred. Reading the ICIVR clears the interrupt code except ARDY(011) RRDY(100) and XRDY(101). Interrupt code for ARDY RRDY and XRDY is cleared when ARDY ICRRDY and ICXRDY bits in the ICSTR is cleared to default value respectively. If other interrupts are pending a new interrupt is generated. If there are more than one interrupt flag reading the ICIVR clears the highest priority interrupt code. Reading the ICIVR also clears corresponding status bit in the ICSTR except ARDY ICRRDY ICXRDY and AAS. Note that users must read (clear) the ICIVR before doing another start otherwise the ICIVR could contain incorrect (old interrupt flags) value. _____ Interrupt Code_____Interrupt Occurred_____ _000_(default)_____None _001_(highest priority)____Arbitration Lost interrupt _010_____No Acknowledgement interrupt _011_____Register Access Ready interrupt _100_____Receive Data Ready interrupt _101_____Transmit Data Ready interrupt _110_____Stop Condition Detection _111_(lowest priority)_____Address As Slave - (RW) _____

25.6.12 ICEMDR Register (Offset = 2Ch) [reset = 0h]

ICEMDR is shown in Figure 25-24 and described in Table 25-16.

Return to Summary Table.

I2C Extended Mode register

Figure 25-24. ICEMDR Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						IGNACK	BCM
R/W-0h						R/W-0h	R/W-0h

Table 25-16. ICEMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved. - (RW)
1	IGNACK	R/W	0h	Ignore NACK mode IGNACK=0 The master transmitter will operate normally discontinue the data transfer and set the ARDY and NACK status bits when a NACK signal is received from the slave. IGNACK=1 The master transmitter will ignore a NACK received from the slave.
0	BCM	R/W	0h	Backward Compatibility Mode. This bit affects the I2C interrupt behavior. Refer to appendix A for details.

25.6.13 ICPSC Register (Offset = 30h) [reset = 0h]

ICPSC is shown in Figure 25-25 and described in Table 25-17.

Return to Summary Table.

I2C Prescaler register

Figure 25-25. ICPSC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU								IPSC7_IPSC0							
R/W-0h								R/W-0h							

Table 25-17. ICPSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved.
7-0	IPSC7_IPSC0	R/W	0h	8-bit prescaler to divide the system clock down to 4/8/12Mhz clock and used by the I2C module. This register must be initialized while the I2C is still in reset (IRS_=0). The value takes effect on the rising edge of IRS_.

25.6.14 ICPID1 Register (Offset = 34h) [reset = 0h]

ICPID1 is shown in Figure 25-26 and described in Table 25-18.

Return to Summary Table.

I2C Peripheral ID register 1

Figure 25-26. ICPID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																CLASS								REVISION							
R/W-0h																R/W-0h								R/W-0h							

Table 25-18. ICPID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved.
15-8	CLASS	R/W	0h	Identifies the class of peripheral. This value should be 0x01 - (RW)
7-0	REVISION	R/W	0h	Identifies the revision level of the I2C. This value should be incremented each time the design is revised. - (RW)

25.6.15 ICPID2 Register (Offset = 38h) [reset = 0h]

ICPID2 is shown in Figure 25-27 and described in Table 25-19.

Return to Summary Table.

I2C Peripheral ID register 2

Figure 25-27. ICPID2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU																								TYPE							
R/W-0h																								R/W-0h							

Table 25-19. ICPID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved.
7-0	TYPE	R/W	0h	Identifies the type of peripheral. This value should be 0x05 - (RW)

25.6.16 ICDMAC Register (Offset = 3Ch) [reset = 0h]

ICDMAC is shown in Figure 25-28 and described in Table 25-20.

Return to Summary Table.

I2C DMA Control Register

Figure 25-28. ICDMAC Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						TXDMAEN	RXDMAEN
R/W-0h						R/W-0h	R/W-0h

Table 25-20. ICDMAC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved. - (RW)
1	TXDMAEN	R/W	0h	Transmit DMA enable. This bit controls the receive DMA event pin to the system. When this bit is 1 the DMA event is enabled and ICTEVT_POR pin is asserted when the DMA transfer is required. When this bit is 0 the ICTEVT_POR pin is never asserted. RXDMAEN=0: DMA transmit event is disabled. RXDMAEN=1: DMA transmit event is enabled. (Default)
0	RXDMAEN	R/W	0h	Receive DMA enable. This bit controls the receive DMA event pin to the system. When this bit is 1 the DMA event is enabled and ICREVT_POR pin is asserted when the DMA transfer is required. When this bit is 0 the ICREVT_POR pin is never asserted. RXDMAEN=0: DMA receive event is disabled. RXDMAEN=1: DMA receive event is enabled. (Default)

25.6.17 ICPFUNC Register (Offset = 48h) [reset = 0h]

ICPFUNC is shown in Figure 25-29 and described in Table 25-21.

Return to Summary Table.

I2C Pin Function register

Figure 25-29. ICPFUNC Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU							PFUNC0
R/W-0h							R/W-0h

Table 25-21. ICPFUNC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R/W	0h	Reserved.
0	PFUNC0	R/W	0h	Controls the function of the I2C SCL and SDA pins. 0 = Pins function as SCL and SDA 1 = Pins functions as GPIO Note: No hardware protection is required to disable I2C function when the PFUNC[0] and IRS_ bits are both set to one. When PFUNC[0] is"1" (GPIO mode) the sub-module which controls the I2C function receives the value"1" for SCL and SDA. IRS_ can be set to"1" regardless of PFUNC[0] and the I2C function works whenever the IRS_ bit is"1". The user is expected to hold I2C in reset via IRS_ bit when changing to/from GPIO mode via the PFUNC[0] bit.

25.6.18 ICPDIR Register (Offset = 4Ch) [reset = 0h]

ICPDIR is shown in Figure 25-30 and described in Table 25-22.

Return to Summary Table.

I2C Pin Direction register

Figure 25-30. ICPDIR Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						PDIR1	PDIR0
R/W-0h						R/W-0h	R/W-0h

Table 25-22. ICPDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDIR1	R/W	0h	Controls the direction of the I2C SDA pin when configured as GPIO. 0 = SDA pin functions as input 1 = SDA pin functions as output
0	PDIR0	R/W	0h	Controls the direction of the I2C SCL pin when configured as GPIO. 0 = SCL pin functions as input 1 = SCL pin functions as output

25.6.19 ICPDIN Register (Offset = 50h) [reset = 0h]

ICPDIN is shown in Figure 25-31 and described in Table 25-23.

Return to Summary Table.

I2C Pin Data In register

Figure 25-31. ICPDIN Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						PDIN1	PDIN0
R/W-0h						R/W-0h	R/W-0h

Table 25-23. ICPDIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDIN1	R/W	0h	Indicates the logic level present on the SDA pin. Reads: 0 = Logic low present at SDA pin regardless of PFUNC setting. 1 = Logic high present at SDA pin regardless of PFUNC setting. Writes: Writes have no effect. - (RW)
0	PDIN0	R/W	0h	Indicates the logic level present on the SCL pin. Reads: 0 = Logic low present at SCL pin regardless of PFUNC setting. 1 = Logic high present at SCL pin regardless of PFUNC setting. Writes: Writes have no effect - (RW)

25.6.20 ICPDOUT Register (Offset = 54h) [reset = 0h]

ICPDOUT is shown in Figure 25-32 and described in Table 25-24.

Return to Summary Table.

I2C Pin Data Out register

Figure 25-32. ICPDOUT Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						PDOUT1	PDOUT0
R/W-0h						R/W-0h	R/W-0h

Table 25-24. ICPDOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDOUT1	R/W	0h	Controls the level driven on the SDA pin when configured as GPIO output. Reads: Reads return register values not GPIO pin levels. Writes: 0 = SDA pin driven low 1 = SDA pin driven high. Note: If SDA is connected to an open-drain buffer at the chiplevel the I2C cannot drive SDA to high.
0	PDOUT0	R/W	0h	Controls the level driven on the SCL pin when configured as GPIO output. Reads: Reads return register values not GPIO pin levels. Writes: 0 = SCL pin driven low 1 = SCL pin driven high Note: If SCL is connected to an open-drain buffer at the chiplevel the I2C cannot drive SCL to high.

25.6.21 ICPDSET Register (Offset = 58h) [reset = 0h]

ICPDSET is shown in Figure 25-33 and described in Table 25-25.

Return to Summary Table.

I2C Pin Data Set register

Figure 25-33. ICPDSET Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						PDSET1	PDSET0
R/W-0h						R/W-0h	R/W-0h

Table 25-25. ICPDSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDSET1	R/W	0h	Used to set PDOUT[1] bit which corresponds to the SDA GPIO pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[1] bit is set to logic high.
0	PDSET0	R/W	0h	Used to set PDOUT[0] bit which corresponds to the SCL GPIO pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[0] bit is set to logic high.

25.6.22 ICPDCLR Register (Offset = 5Ch) [reset = 0h]

ICPDCLR is shown in Figure 25-34 and described in Table 25-26.

Return to Summary Table.

I2C Pin Data Clear register

Figure 25-34. ICPDCLR Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						PDCLR1	PDCLR0
R/W-0h						R/W-0h	R/W-0h

Table 25-26. ICPDCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDCLR1	R/W	0h	Used to clear PDOUT[1] bit which corresponds to the SDA pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[1] bit is cleared to logic low.
0	PDCLR0	R/W	0h	Used to clear PDOUT[0] bit which corresponds to the SCL pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[0] bit is cleared to logic low.

25.6.23 ICPDRV Register (Offset = 60h) [reset = 0h]

ICPDRV is shown in Figure 25-35 and described in Table 25-27.

Return to Summary Table.

I2C Pin Driver Mode Register

Figure 25-35. ICPDRV Register

31	30	29	28	27	26	25	24
NU							
R/W-0h							
23	22	21	20	19	18	17	16
NU							
R/W-0h							
15	14	13	12	11	10	9	8
NU							
R/W-0h							
7	6	5	4	3	2	1	0
NU						PDRV1	PDRV0
R/W-0h						R/W-0h	R/W-0h

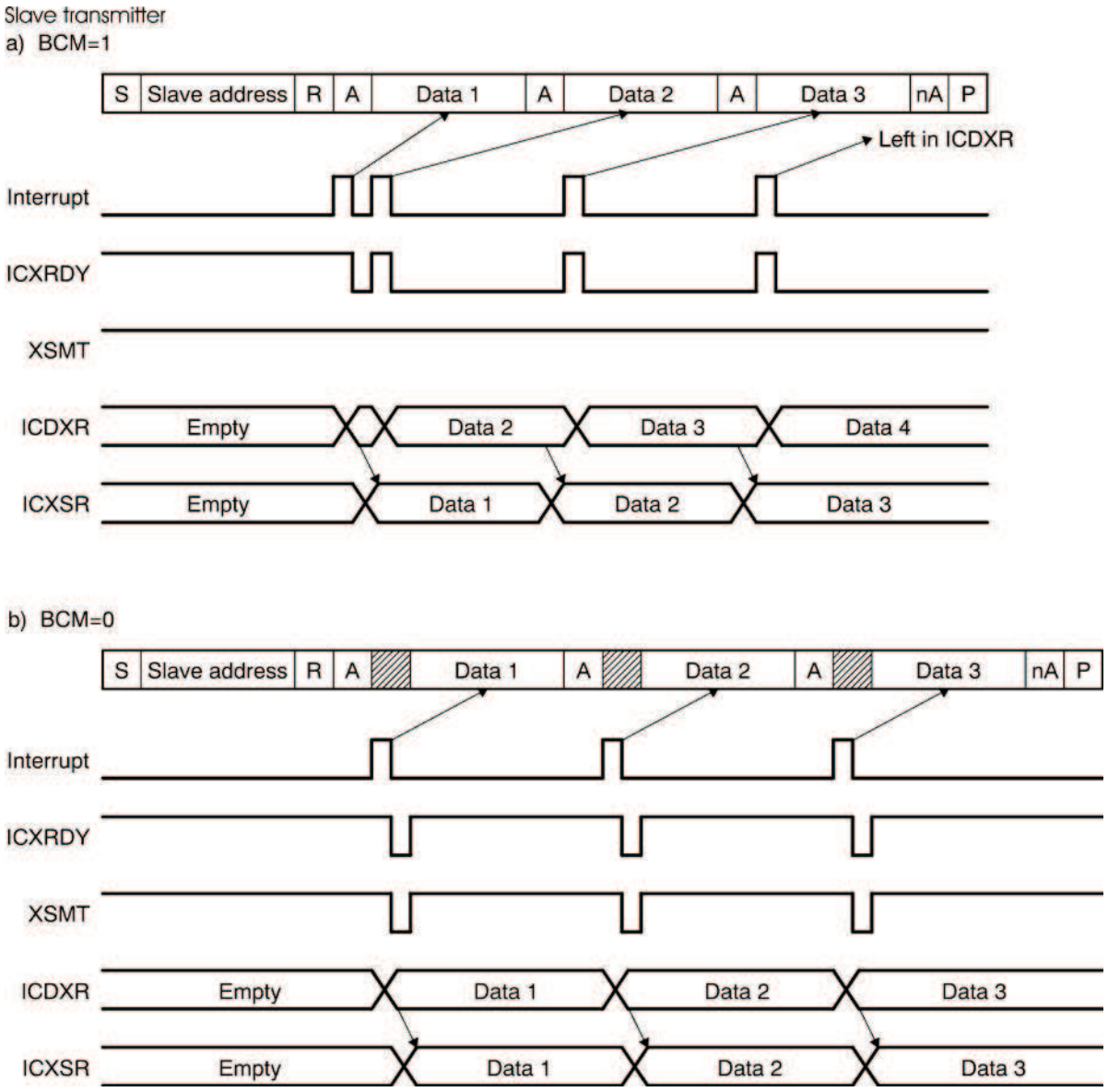
Table 25-27. ICPDRV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDRV1	R/W	0h	Used to select driver mode of output buffer for SDA pin. 0 = I2C mode. 1 = GPIO mode. Note: Value of this register is reflected on the PDRV_SDA_POR port. Actual function depends on I/O buffer and chip implementation.
0	PDRV0	R/W	0h	Used to select driver mode of output buffer for SCL pin. 0 = I2C mode. 1 = GPIO mode. Note: Value of this register is reflected on the PDRV_SCL_POR port. Actual function depends on I/O buffer and chip implementation.

25.7 Sample Waveforms

Figure 25-36 provides waveforms to illustrate the difference between normal operation and backward compatibility mode.

Figure 25-36. Difference between Normal Operation and Backward Compatibility Mode



Serial Communication Interface (SCI)

This section contains the description of the serial communication interface (SCI).

Topic **Page**

26.1 Introduction **3235**
26.2 SCI Communication Formats **3237**
26.3 SCI Interrupts **3242**
26.4 SCI DMA Interface **3245**
26.5 SCI Configurations **3246**
26.6 SCI Low Power Mode **3248**
26.7 SCI Control Registers **3250**
26.8 GPIO Functionality **3280**

26.1 Introduction

The SCI module is a universal asynchronous receiver-transmitter that implements the standard nonreturn to zero format. The SCI can be used to communicate, for example, through an RS-232 port or over a K-line.

26.1.1 SCI Features

The following are the features of the SCI module:

- Standard universal asynchronous receiver-transmitter (UART) communication
- Supports full- or half-duplex operation
- Standard nonreturn to zero (NRZ) format
- Double-buffered receive and transmit functions
- Supports two individually enabled interrupt lines: level 0 and level 1
- Configurable frame format of 3 to 13 bits per character based on the following:
  - Data word length programmable from one to eight bits
  - Additional address bit in address-bit mode
  - Parity programmable for zero or one parity bit, odd or even parity
  - Stop programmable for one or two stop bits
- Asynchronous or isosynchronous communication modes with no CLK pin
- Two multiprocessor communication formats allow communication between more than two devices
- Sleep mode is available to free CPU resources during multiprocessor communication and then wake up to receive an incoming message
- The 24-bit programmable baud rate supports 2^{24} different baud rates provide high accuracy baud rate selection
- Capability to use Direct Memory Access (DMA) for transmit and receive data
- Four error flags and Five status flags provide detailed information regarding SCI events
- Two external pins: SCIRX and SCITX

NOTE: SCI module does not support UART Hardware Flow Control. This feature can be implemented in Software using a General Purpose I/O pin.

26.1.2 Block Diagram

Three Major components of the SCI Module are:

- Transmitter
- Baud Clock Generator
- Receiver

Transmitter (TX) contains two major registers to perform double buffering:

- The transmitter data buffer register (SCITD) contains data loaded by the CPU to be transferred to the shift register for transmission.
- The transmitter shift register (SCITXSHF) loads data from the data buffer (SCITD) and shifts data onto the SCITX pin, one bit at a time.

Baud Clock Generator

- A programmable baud generator produces a baud clock scaled from VBUSP CLK.

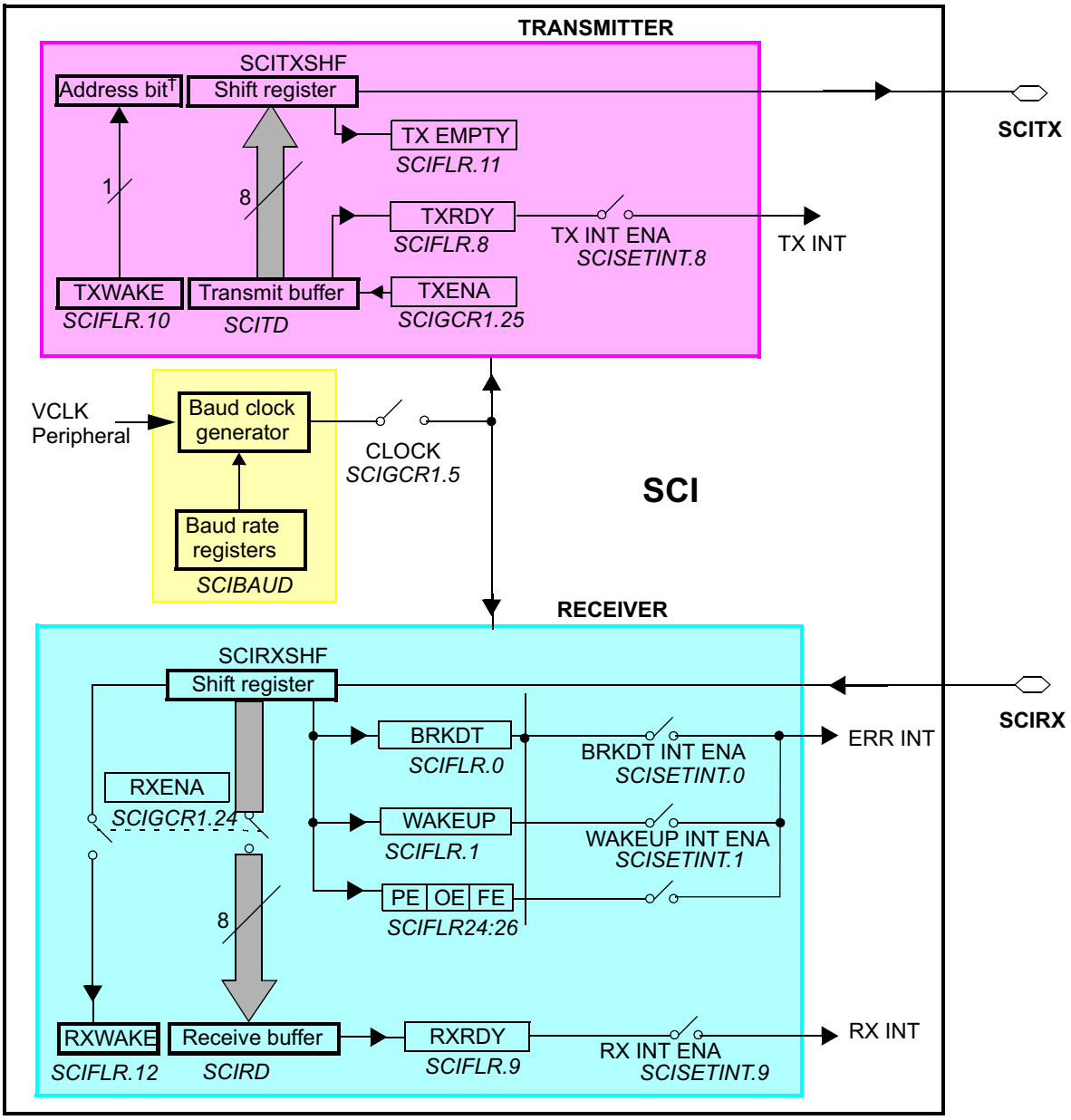
Receiver (RX) contains two major registers to perform double buffering:

- The receiver shift register (SCIRXSHF) shifts data in from the SCIRX pin one bit at a time and transfers completed data into the receive data buffer.
- The receiver data buffer register (SCIRD) contains received data transferred from the receiver shift register

The SCI receiver and transmitter are double-buffered, and each has its own separate enable and interrupt bits. The receiver and transmitter may each be operated independently or simultaneously in full duplex mode.

To ensure data integrity, the SCI checks the data it receives for breaks, parity, overrun, and framing errors. The bit rate (baud) is programmable to over 16 million different rates through a 24-bit baud-select register. Figure 26-1 shows the detailed SCI block diagram.

Figure 26-1. Detailed SCI Block Diagram



26.2 SCI Communication Formats

The SCI module can be configured to meet the requirements of many applications. Because communication formats vary depending on the specific application, many attributes of the SCI are user configurable. The list below describes these configuration options:

- SCI Frame format
- SCI Timing modes
- SCI Baud rate
- SCI Multiprocessor modes

26.2.1 SCI Frame Formats

The SCI uses a programmable frame format. All frames consist of the following:

- One start bit
- One to eight data bits
- Zero or one address bit
- Zero or one parity bit
- One or two stop bits

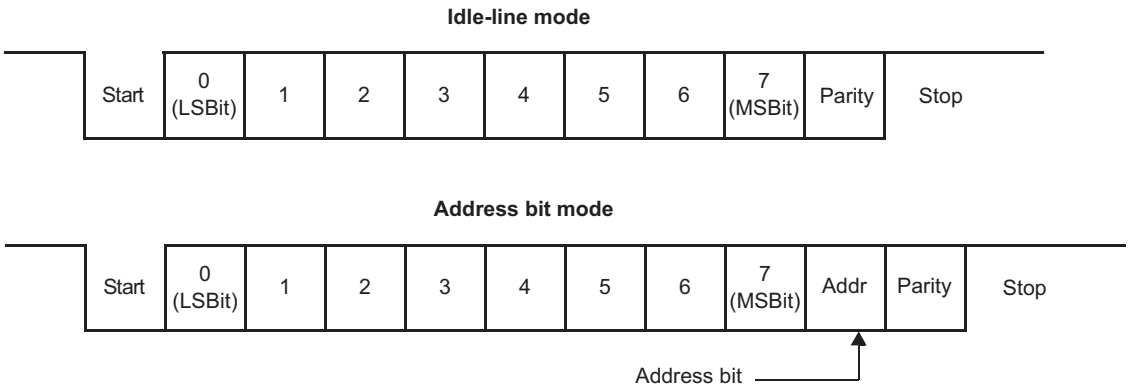
The frame format for both the transmitter and receiver is programmable through the bits in the SCIGCR1 register. Both receive and transmit data is in nonreturn to zero (NRZ) format, which means that the transmit and receive lines are at logic high when idle. Each frame transmission begins with a start bit, in which the transmitter pulls the SCI line low (logic low). Following the start bit, the frame data is sent and received least significant bit first (LSB).

An address bit is present in each frame if the SCI is configured to be in address-bit mode but is not present in any frame if the SCI is configured for idle-line mode. The format of frames with and without the address bit is illustrated in Figure 26-2.

A parity bit is present in every frame when the PARITY ENA bit is set. The value of the parity bit depends on the number of one bits in the frame and whether odd or even parity has been selected via the PARITY ENA bit. Both examples in Figure 26-2 have parity enabled.

All frames include one stop bit, which is always a high level. This high level at the end of each frame is used to indicate the end of a frame to ensure synchronization between communicating devices. Two stop bits are transmitted if the STOP bit in SCIGCR1 register is set. The examples shown in Figure 26-2 use one stop bit per frame.

Figure 26-2. Typical SCI Data Frame Formats



26.2.2 SCI Timing Mode

The SCI can be configured to use asynchronous or isosynchronous timing using TIMING MODE bit in SCIGCR1 register.

26.2.2.1 Asynchronous Timing Mode

The asynchronous timing mode uses only the receive and transmit data lines to interface with devices using the standard universal asynchronous receiver- transmitter (UART) protocol.

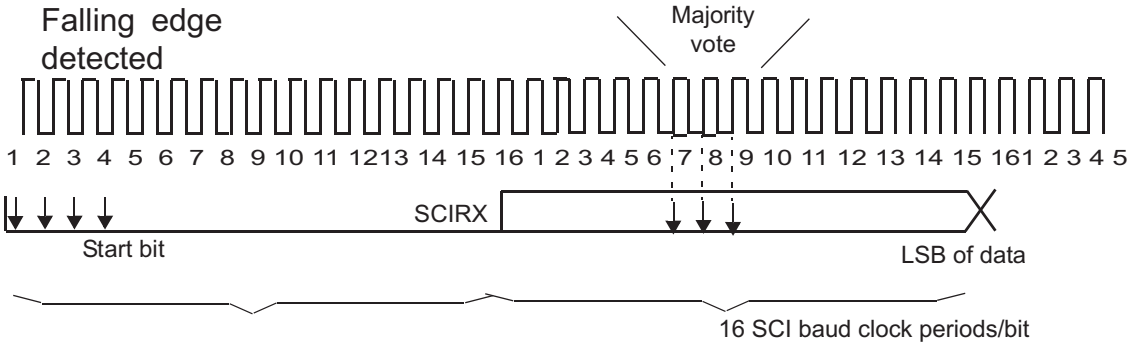
In the asynchronous timing mode, each bit in a frame has a duration of 16 SCI baud clock periods. Each bit therefore consists of 16 samples (one for each clock period). When the SCI is using asynchronous mode, the baud rates of all communicating devices must match as closely as possible. Receive errors result from devices communicating at different baud rates.

With the receiver in the asynchronous timing mode, the SCI detects a valid start bit if the first four samples after a falling edge on the SCIRX pin are of logic level 0. As soon as a falling edge is detected on SCIRX, the SCI assumes that a frame is being received and synchronizes itself to the bus.

To prevent interpreting noise as Start bit SCI expects SCIRX line to be low for at least four contiguous SCI baud clock periods to detect a valid start bit. The bus is considered idle if this condition is not met. When a valid start bit is detected, the SCI determines the value of each bit by sampling the SCIRX line value during the seventh, eighth, and ninth SCI baud clock periods. A majority vote of these three samples is used to determine the value stored in the SCI receiver shift register. By sampling in the middle of the bit, the SCI reduces errors caused by propagation delays and rise and fall times and data line noises. Figure 26-3 illustrates how the receiver samples a start bit and a data bit in asynchronous timing mode.

The transmitter transmits each bit for a duration of 16 SCI baud clock periods. During the first clock period for a bit, the transmitter shifts the value of that bit onto the SCITX pin. The transmitter then holds the current bit value on SCITX for 16 SCI baud clock periods.

Figure 26-3. Asynchronous Communication Bit Timing



26.2.2.2 Isosynchronous Timing Mode

In isosynchronous timing mode, each bit in a frame has a duration of exactly 1 baud clock period and therefore consists of a single sample. With this timing configuration, the transmitter and receiver are required to make use of the SCICLK pin to synchronize communication with other SCI. **This mode is not fully supported on this device because SCICLK pin is not available.**

26.2.3 SCI Baud Rate

The SCI has an internally generated serial clock determined by the peripheral VCLK and the prescalers BAUD. The SCI uses the 24-bit integer prescaler BAUD value in the BRS register to select the required baud rates.

In asynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$\textit{Asynchronous baud value} = \frac{\text{VBUSPCLK Frequency}}{16\ \text{BAUD} + 1}$$

For BAUD = 0,

$$\textit{Asynchronous baud value} = \frac{\text{VCLK Frequency}}{32} \quad (15)$$

In isosynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$\textit{Isosynchronous baud value} = \frac{\text{VBUSPCLK Frequency}}{\text{BAUD} + 1}$$

For BAUD = 0,

$$\textit{Isosynchronous baud value} = \frac{\text{VCLK Frequency}}{32} \quad (16)$$

26.2.4 SCI Multiprocessor Communication Modes

In some applications, the SCI may be connected to more than one serial communication device. In such a multiprocessor configuration, several frames of data may be sent to all connected devices or to an individual device. In the case of data sent to an individual device, the receiving devices must determine when they are being addressed. When a message is not intended for them, the devices can ignore the following data. When only two devices make up the SCI network, addressing is not needed, so multiprocessor communication schemes are not required.

SCI supports two multiprocessor Communication Modes which can be selected using COMM MODE bit:

- Idle-Line Mode
- Address Bit Mode

When the SCI is not used in a multiprocessor environment, software can consider all frames as data frames. In this case, the only distinction between the idle-line and address-bit modes is the presence of an extra bit (the address bit) in each frame sent with the address-bit protocol.

The SCI allows full-duplex communication where data can be sent and received via the transmit and receive pins simultaneously. However, the protocol used by the SCI assumes that only one device transmits data on the same bus line at any one time. No arbitration is done by the SCI.

NOTE: Avoid Transmitting Simultaneously on the Same Serial Bus

The system designer must ensure that devices connected to the same serial bus line do not attempt to transmit simultaneously. If two devices are transmitting different data, the resulting bus conflict could damage the device..

26.2.4.1 Idle-Line Multiprocessor Modes

In idle-line multiprocessor mode, a frame that is preceded by an idle period (10 or more idle bits) is an address frame. A frame that is preceded by fewer than 10 idle bits is a data frame. Figure 26-4 illustrates the format of several blocks and frames with idle-line mode.

There are two ways to transmit an address frame using idle-line mode:

Method 1: In software, deliberately leave an idle period between the transmission of the last data frame of the previous block and the address frame of the new block.

Method 2: Configure the SCI to automatically send an idle period between the last data frame of the previous block and the address frame of the new block.

Although Method 1 is only accomplished by a delay loop in software, Method 2 can be implemented by using the transmit buffer and the TXWAKE bit in the following manner:

Step1 : Write a 1 to the TXWAKE bit.

Step2 : Write a dummy data value to the SCITD register. This triggers the SCI to begin the idle period as soon as the transmitter shift register is empty.

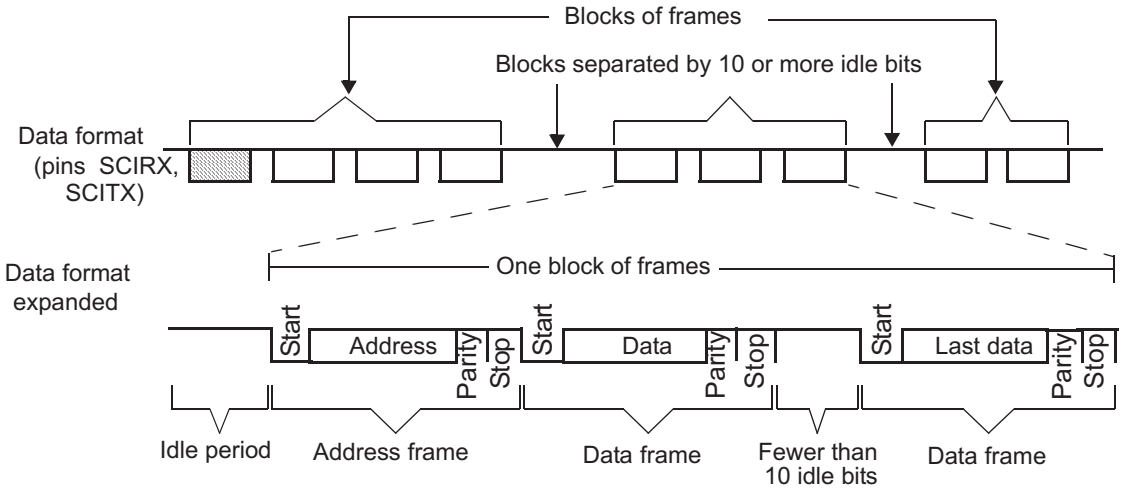
Step3 : Wait for the SCI to clear the TXWAKE flag.

Step4 : Write the address value to SCITD.

As indicated by Step 3, software should wait for the SCI to clear the TXWAKE bit. However, the SCI clears the TXWAKE bit at the same time it sets TXRDY (that is, transfers data from SCITD into SCITXSHF). Therefore, if the TX INT ENA bit is set, the transfer of data from SCITD to SCITXSHF causes an interrupt to be generated at the same time that the SCI clears the TXWAKE bit. If this interrupt method is used, software is not required to poll the TXWAKE bit waiting for the SCI to clear it.

When idle-line multiprocessor communications are used, software must ensure that the idle time exceeds 10 bit periods before addresses (using one of the methods mentioned above), and software must also ensure that data frames are written to the transmitter quickly enough to be sent without a delay of 10 bit periods between frames. Failure to comply with these conditions will result in data interpretation errors by other devices receiving the transmission.

Figure 26-4. Idle-Line Multiprocessor Communication Format



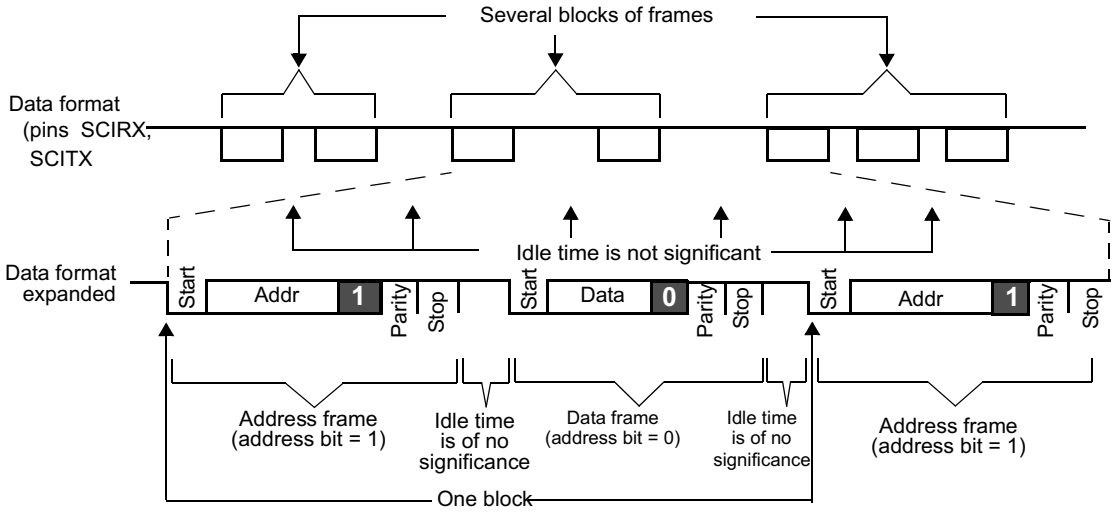
26.2.4.2 Address-Bit Multiprocessor Mode

In the address-bit protocol, each frame has an extra bit immediately following the data field called an address bit. A frame with the address bit set to 1 is an address frame; a frame with the address bit set to 0 is a data frame. The idle period timing is irrelevant in this mode. Figure 26-5 illustrates the format of several blocks and frames with the address-bit mode.

When address-bit mode is used, the value of the TXWAKE bit is the value sent as the address bit. To send an address frame, software must set the TXWAKE bit. This bit is cleared as the contents of the SCITD are shifted from the TXWAKE register so that all frames sent are data except when the TXWAKE bit is written as a 1.

No dummy write to SCITD is required before an address frame is sent in address-bit mode. The first byte written to SCITD after the TXWAKE bit is written to 1 is transmitted with the address bit set when address-bit mode is used.

Figure 26-5. Address-Bit Multiprocessor Communication Format



26.3 SCI Interrupts

The SCI module has two interrupt lines, level 0 and level 1, to the vectored interrupt manager (VIM) module (see Figure 26-6). Two offset registers SCIINTVECT0 and SCIINTVECT1 determine which flag triggered the interrupt according to the respective priority encoders. Each interrupt condition has a bit to enable/disable the interrupt in the SCISETINT and SCICLRINT registers, respectively.

Each interrupt also has a bit that can be set as interrupt level 0 (INT0) or as interrupt level 1 (INT1). By default, interrupts are in interrupt level 0. SCISETINTLVL sets a given interrupt to level1. SCICLEARINTLVL resets a given interrupt level to the default level 0.

The interrupt vector registers SCIINTVECT0 and SCIINTVECT1 return the vector of the pending interrupt line INT0 or INT1. If more than one interrupt is pending, the interrupt vector register holds the highest priority interrupt.

Figure 26-6. General Interrupt Scheme

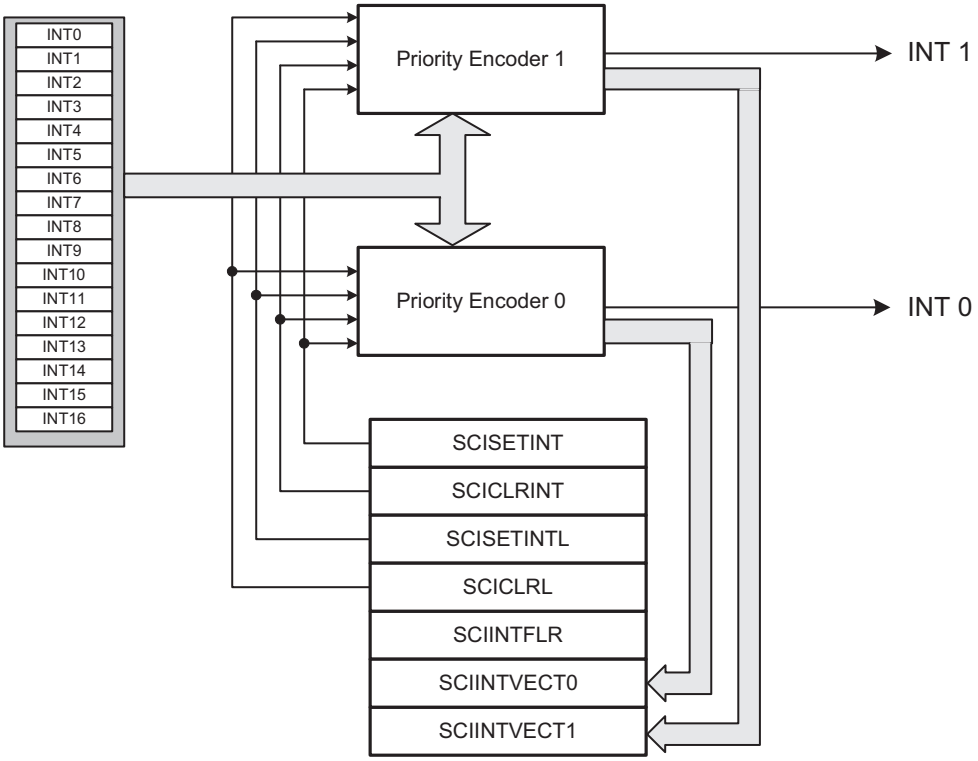
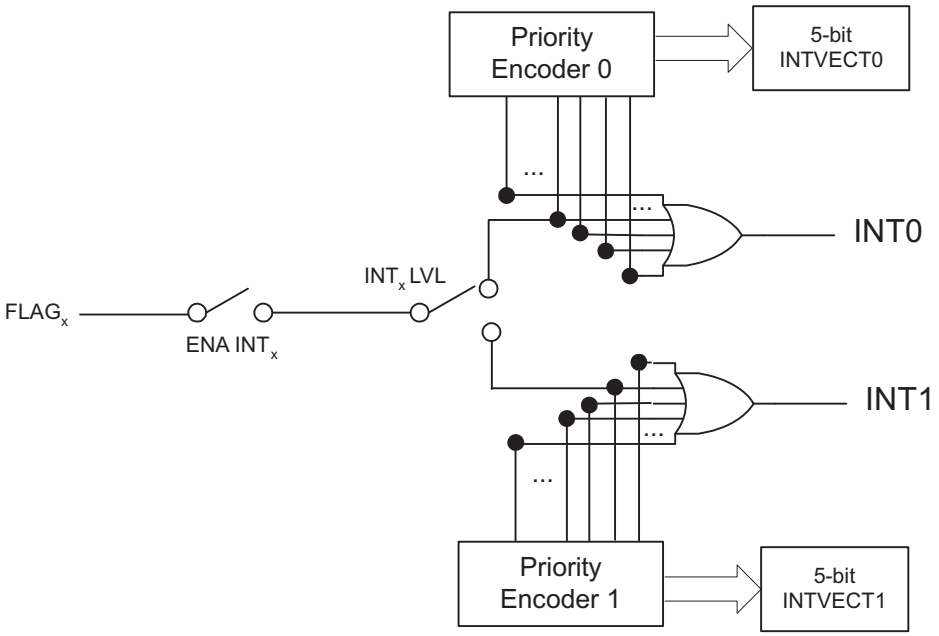


Figure 26-7. Interrupt Generation for Given Flags



26.3.1 Transmit Interrupt

To use transmit interrupt functionality, SET TX INT bit must be enabled and SET TX DMA bit must be cleared. The transmit ready (TXRDY) flag is set when the SCI transfers the contents of SCITD to the shift register, SCITXSHF. The TXRDY flag indicates that SCITD is ready to be loaded with more data. In addition, the SCI sets the TX EMPTY bit if both the SCITD and SCITXSHF registers are empty. If the SET TX INT bit is set, then a transmit interrupt is generated when the TXRDY flag goes high. Transmit Interrupt is not generated immediately after setting the SET TX INT bit unlike transmit DMA request. Transmit Interrupt is generated only after the first transfer from SCITD to SCITXSHF, that is first data has to be written to SCITD by the User before any interrupt gets generated. To transmit further data the user can write data to SCITD in the transmit Interrupt service routine.

Writing data to the SCITD register clears the TXRDY bit. When this data has been moved to the SCITXSHF register, the TXRDY bit is set again. The interrupt request can be suspended by setting the CLR TX INT bit; however, when the SET TX INT bit is again set to 1, the TXRDY interrupt is asserted again. The transmit interrupt request can be eliminated until the next series of values is written to SCITD, by disabling the transmitter via the TXENA bit, by a software reset SWnRST, or by a device hardware reset.

26.3.2 Receive Interrupt

The receive ready (RXRDY) flag is set when the SCI transfers newly received data from SCIRXSHF to SCIRD. The RXRDY flag therefore indicates that the SCI has new data to be read. Receive interrupts are enabled by the SET RX INT bit. If the SET RX INT is set when the SCI sets the RXRDY flag, then a receive interrupt is generated. The received data can be read in the Interrupt Service routine.

On a device with both SCI and a DMA controller, the bits SET RX DMA ALL and SET RX DMA must be cleared to select interrupt functionality.

26.3.3 WakeUp Interrupt

SCI sets the WAKEUP flag if bus activity on the RX line either prevents power-down mode from being entered, or RX line activity causes an exit from power-down mode. If enabled (SET WAKEUP INT), wakeup interupt is triggered once WAKEUP flag is set.

26.3.4 Error Interrupts

The following error detection features are supported with Interrupt by the SCI module:

- Parity errors (PE)
- Frame errors (FE)
- Break Detect errors (BRKDT)
- Overrun errors (OE)

If any of these errors (PE, FE, BRKDT, OE) is flagged, an interrupt for the flagged errors will be generated if enabled. A message is valid for both the transmitter and the receiver if there is no error detected until the end of the frame. Each of these flags is located in the receiver status (SCIFLR) register. Further details on these flags are explained in SCIFLR register description.

The SCI module supports following 7 interrupts as seen in Table 26-1.

Table 26-1. SCI Interrupts

Offset [1]	Interrupt
0	Reserved
1	Wakeup
2	Reserved
3	Parity error
4	Reserved
5	Reserved
6	Frame error
7	Break detect error
8	Reserved
9	Overrun error
10	Reserved
11	Receive
12	Transmit
13 - 15	Reserved

(1) Offset 1 is the highest priority. Offset 16 is the lowest priority.

26.4 SCI DMA Interface

DMA requests for receive (RXDMA request) and transmit (TXDMA request) are available for the SCI module. Refer to the DMA module chapter for DMA module configurations.

26.4.1 Receive DMA Requests

This DMA functionality is enabled/disabled by the CPU using the SET RX DMA/CLR RX DMA bits, respectively.

The receiver DMA request is set when a frame is received successfully and DMA functionality has been previously enabled. The RXRDY flag is set when the SCI transfers newly received data from the SCIRXSHF register to the SCIRD buffer. The RXRDY flag therefore indicates that the SCI has new data to be read. Receive DMA requests are enabled by the SET RX INT bit.

Parity, overrun, break detect, wake-up, and framing errors generate an error interrupt request immediately upon detection, if enabled, even if the device is in the process of a DMA data transfer. The DMA transfer is postponed until the error interrupt is served. The error interrupt can delete this particular DMA request by reading the receive buffer.

In multiprocessor mode, the SCI can generate receiver interrupts for address frames and DMA requests for data frames. This is controlled by an extra select bit SET RX DMA ALL.

If the SET RX DMA ALL bit is set and the SET RX DMA bit is set when the SCI sets the RXRDY flag, then a receive DMA request is generated for address and data frames.

If the SET RX DMA ALL bit is cleared and the SET RX DMA bit is set when the SCI sets the RXRDY flag upon receipt of a data frame, then a receive DMA request is generated. Receive interrupt requests are generated for address frames.

In multiprocessor mode with the SLEEP bit set, no DMA is generated for received data frames. The software must clear the SLEEP bit before data frames can be received. Table 26-2 specifies the bit values for DMA requests in multiprocessor modes.

In multiprocessor mode, the SCI can generate receiver interrupts for address frames and DMA requests for data frames or DMA requests for both. This is controlled by the SET RX DMA ALL bit.

In multiprocessor mode with the SLEEP bit set, no DMA is generated for received data frames. The software must clear the SLEEP bit before data frames can be received.

Table 26-2. DMA and Interrupt Requests in Multiprocessor Modes

SET RX INT	SET RX DMA	SET RX DMA ALL	ADDR FRAME INT	ADDR FRAME DMA	DATA FRAME INT	DATA FRAME DMA
0	0	x	N	N	N	N
0	1	0	Y	N	N	Y
0	1	1	N	Y	N	Y
1	0	x	Y	N	Y	N
1	1	0	Y	N	Y	Y
1	1	1	Y	Y	Y	Y

26.4.2 Transmit DMA Requests

DMA functionality is enabled/disabled by the CPU with SET TX DMA/CLR TX DMA bits, respectively.

The TXRDY flag is set when the SCI transfers the contents of SCITD to SCITXSHF. The TXRDY flag indicates that SCITD is ready to be loaded with more data. In addition, the SCI sets the TX EMPTY bit if both the SCITD and SCITXSHF registers are empty.

Transmit DMA requests are enabled by the setting SET TX DMA and SET TX INT bits. If the SET TX DMA bit is set, then a TX DMA request is sent to the DMA when data is written to SCITD and TXRDY is set. In other words, CPU needs to write the first data to start a DMA block transfer. For example, we want to transmit a data buffer of 20 bytes. DMA will be set up to transmit 19 bytes. The first data for DMA to transfer is the second byte in the buffer. CPU will have to write the first byte in the buffer to the SCITD register to start the transfer.

.

26.5 SCI Configurations

Before the SCI sends or receives data, its registers should be properly configured. Upon power-up or a system-level reset, each bit in the SCI registers is set to a default state. The registers are writable only after the RESET bit is set to 1. Of particular importance is the SWnRST bit. This active-low bit is initialized to 0 and keeps the SCI in a reset state until it is programmed to 1. Therefore, all SCI configuration should be completed before a 1 (one) is written to the SWnRST bit.

The following list details the configuration steps that software should perform prior to the transmission or reception of data. As long as SWnRST is held low the entire time that the SCI is being configured, the order in which the registers are programmed is not important.

- Enable SCI by setting RESET bit.
- Clear SWnRST to 0 before configuring the SCI.
- Select the desired frame format by programming SCIGCR1.
- Configure the SCIRX and SCITX pins for SCI functionality by setting the RX FUNC and TX FUNC bit.
- Select the baud rate to be used for communication by programming BRS.
- Select internal clock by programming the CLOCK bit.
- Set the CONT bit to make SCI not to halt for an emulation breakpoint until its current reception or transmission is complete (this bit is used only in an emulation environment).
- Set LOOP BACK bit to connect the transmitter to the receiver internally (this feature is used to perform a self-test).
- Select the receiver enable RXENA bit if data is to be received.
- Select the transmit enable TXENA bit if data is to be transmitted.
- Set SWnRST to 1 after the SCI is configured.
- Perform Receive or Transmit data (see Section 26.5.1 and Section 26.5.2).

26.5.1 Receiving Data

The SCI receiver is enabled to receive messages if the RX FUNC bit and the RXENA bit are set to 1. If the RX FUNC bit is not set, the SCIRX pin functions as a general purpose I/O pin rather than as an SCI function pin. After a valid idle period is detected, data is automatically received as it arrives on the SCIRX pin.

SCI sets the RXRDY bit when it transfers newly received data from SCIRXSHF to SCIRD. The SCI clears the RXRDY bit after the new data in SCIRD has been read. Also, as data is transferred from SCIRXSHF to SCIRD, the SCI sets FE, OE, or PE if any of these error conditions were detected in the received data. These error conditions are supported with configurable Interrupt capability. The wake-up and break-detect status bits are also set if one of these errors occurs, but they do not necessarily occur at the same time that new data is being loaded into SCIRD.

User can receive data by:

1. Polling Receive Ready Flag
2. Receive Interrupt
3. DMA

In polling method, software can poll for RXRDY bit and read the data from SCIRD register once RXRDY is set high. CPU is unnecessarily overloaded by selecting Polling mode. To avoid this user can use either Interrupt or DMA method. To use interrupt method SET RX INT bit should be set and to use DMA SET RX DMA bit should be set. Either an Interrupt or a DMA request is generated the moment RXRDY is set.

26.5.2 Transmitting Data

The SCI transmitter is enabled if the TX FUNC bit and the TXENA bit are set to 1. If the TX FUNC bit is not set, the SCITX pin functions as a general purpose I/O pin rather than as an SCI function pin. Any value written to the SCITD before TXENA is set to 1 is not transmitted. Both of these control bits allow for the SCI transmitter to be held inactive independently of the receiver.

SCI waits for data to be written to SCITD, transfers it to SCITXSHF, and transmits it. The flags TXRDY and TX EMPTY indicate the status of the transmit buffers. That is, when the transmitter is ready for data to be written to SCITD, the TXRDY bit is set. Additionally, if both SCITD and SCITXSHF are empty, then the TX EMPTY bit is also set.

User can transmit data by:

1. Polling Transmit Ready Flag
2. Receive Interrupt
3. DMA

In polling method, software can poll for TXRDY bit to go high before writing the data to SCITD register. CPU is unnecessarily overloaded by doing this Polling method. To avoid this user can use either Interrupt or DMA method. To use interrupt method SET TX INT bit should be set and to use DMA SET TX DMA bit should be set. Either an Interrupt or a DMA request is generated the moment TXRDY is set. When the SCI has completed transmission of all pending frames, the SCITXSHF register and SCITD are empty, the TXRDY bit is set, and an interrupt/DMA request is generated, if enabled. Because all data has been transmitted, the interrupt/DMA request should be halted. This can be done by either disabling the transmit interrupt (CLR TX INT) / DMA request (CLR TX DMA bit) or by disabling the transmitter (clear TXENA bit).

NOTE: The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0 or SCIINTVECT1 register.

26.6 SCI Low Power Mode

The SCI can be put in either local or global low-power mode. Global low-power mode is asserted by the system and is not controlled by the SCI. During global low-power mode, all clocks to the SCI are turned off so the module is completely inactive.

Local low-power mode is asserted by setting the POWERDOWN bit; setting this bit stops the clocks to the SCI internal logic and the module registers. Setting the POWERDOWN bit causes the SCI to enter local low-power mode and clearing the POWERDOWN bit causes SCI to exit from local low-power mode. All the registers are accessible during local power-down mode as any register access enables the clock to SCI for that particular access alone.

The wake-up interrupt is used to allow the SCI to exit low-power mode automatically when a low level is detected on the SCIRX pin and also this clears the POWERDOWN bit. If wake-up interrupt is disabled, then the SCI immediately enters low-power mode whenever it is requested and also any activity on the SCIRX pin does not cause the SCI to exit low-power mode.

NOTE: Enabling Local Low-Power Mode During Receive and Transmit

If the wake-up interrupt is enabled and low-power mode is requested while the receiver is receiving data, then the SCI immediately generates a wake-up interrupt to clear the powerdown bit and prevents the SCI from entering low-power mode and thus completes the current reception. Otherwise, if the wake-up interrupt is disabled, then the SCI completes the current reception and then enters the low-power mode.

26.6.1 Sleep Mode for Multiprocessor Communication

When the SCI receives data and transfers that data from SCIRXSHF to SCIRD, the RXRDY bit is set and if RX INT ENA is set, the SCI also generates an interrupt. The interrupt triggers the CPU to read the newly received frame before another one is received. In multiprocessor communication modes, this default behavior may be enhanced to provide selective indication of new data. When SCI receives an address frame that does not match its address, the device can ignore the data following this non-matching address until the next address frame by using sleep mode. Sleep mode can be used with both idle-line and address-bit multiprocessor modes.

If sleep mode is enabled by the SLEEP bit, then the SCI transfers data from SCIRXSHF to SCIRD only for address frames. Therefore, in sleep mode, all data frames are assembled in the SCIRXSHF register without being shifted into the SCIRD and without initiating a receive interrupt or DMA request. Upon reception of an address frame, the contents of the SCIRXSHF are moved into SCIRD, and the software must read SCIRD and determine if the SCI is being addressed by comparing the received address against the address previously set in the software and stored somewhere in memory (the SCI does not have hardware available for address comparison). If the SCI is being addressed, the software must clear the SLEEP bit so that the SCI will load SCIRD with the data of the data frames that follow the address frame.

When the SCI has been addressed and sleep mode has been disabled (in software) to allow the receipt of data, the SCI should check the RXWAKE bit (SCIFLR.12) to determine when the next address has been received. This bit is set to 1 if the current value in SCIRD is an address and set to 0 if SCIRD contains data. If the RXWAKE bit is set, then software should check the address in SCIRD against its own address. If it is still being addressed, then sleep mode should remain disabled. Otherwise, the SLEEP bit should be set again.

Following is a sequence of events typical of sleep mode operation:

- The SCI is configured and both sleep mode and receive actions are enabled.
- An address frame is received and a receive interrupt is generated.
- Software compares the received address frame against that set by software and determines that the SCI is not being addressed, so the value of the SLEEP bit is not changed.
- Several data frames are shifted into SCIRXSHF, but no data is moved to SCIRD and no receive interrupts are generated.
- A new address frame is received and a receive interrupt is generated.
- Software compares the received address frame against that set by software and determines that the SCI is being addressed and clears the SLEEP bit.
- Data shifted into SCIRXSHF is transferred to SCIRD, and a receive interrupt is generated after each data frame is received.
- In each interrupt routine, software checks RXWAKE to determine if the current frame is an address frame.
- Another address frame is received, RXWAKE is set, software determines that the SCI is not being addressed and sets the SLEEP bit back to 1. No receive interrupts are generated for the data frames following this address frame.

By ignoring data frames that are not intended for the device, fewer interrupts are generated. These interrupts would otherwise require CPU intervention to read data that is of no significance to this specific device. Using sleep mode can help free some CPU resources.

Except for the RXRDY flag, the SCI continues to update the receiver status flags (see Table 26-11) while sleep mode is active. In this way, if an error occurs on the receive line, an application can immediately respond to the error and take the appropriate corrective action.

Because the RXRDY bit is not updated for data frames when sleep mode is enabled, the SCI can enable sleep mode and use a polling algorithm if desired. In this case, when RXRDY is set, software knows that a new address has been received. If the SCI is not being addressed, then the software should not change the value of the SLEEP bit and should continue to poll RXRDY.

26.7 SCI Control Registers

These registers are accessible in 8-, 16-, and 32-bit reads or writes. The SCI is controlled and accessed through the registers listed in Table 26-3. Among the features that can be programmed are the SCI communication and timing modes, baud rate value, frame format, DMA requests, and interrupt configuration. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

Table 26-3. SCI Control Registers Summary

Offset	Acronym	Register Description	Section
00h	SCIGCR0	SCI Global Control Register 0	Section 26.7.1
04h	SCIGCR1	SCI Global Control Register 1	Section 26.7.2
0Ch	SCISETINT	SCI Set Interrupt Register	Section 26.7.3
10h	SCICLEARINT	SCI Clear Interrupt Register	Section 26.7.4
14h	SCISETINTLVL	SCI Set Interrupt Level Register	Section 26.7.5
18h	SCICLEARINTLVL	SCI Clear Interrupt Level Register	Section 26.7.6
1Ch	SCIFLR	SCI Flags Register	Section 26.7.7
20h	SCIINTVECT0	SCI Interrupt Vector Offset 0	Section 26.7.8
24h	SCIINTVECT1	SCI Interrupt Vector Offset 1	Section 26.7.9
28h	SCIFORMAT	SCI Format Control Register	Section 26.7.10
2Ch	BRS	Baud Rate Selection Register	Section 26.7.11
30h	SCIED	Receiver Emulation Data Buffer	Section 26.7.12.1
34h	SCIRD	Receiver Data Buffer	Section 26.7.12.2
38h	SCITD	Transmit Data Buffer	Section 26.7.12.3
3Ch	SCIPIO0	SCI Pin I/O Control Register 0	Section 26.7.13
40h	SCIPIO1	SCI Pin I/O Control Register 1	Section 26.7.14
44h	SCIPIO2	SCI Pin I/O Control Register 2	Section 26.7.15
48h	SCIPIO3	SCI Pin I/O Control Register 3	Section 26.7.16
4Ch	SCIPIO4	SCI Pin I/O Control Register 4	Section 26.7.17
50h	SCIPIO5	SCI Pin I/O Control Register 5	Section 26.7.18
54h	SCIPIO6	SCI Pin I/O Control Register 6	Section 26.7.19
58h	SCIPIO7	SCI Pin I/O Control Register 7	Section 26.7.20
5Ch	SCIPIO8	SCI Pin I/O Control Register 8	Section 26.7.21
90h	IODFTCTRL	Input/Output Error Enable Register	Section 26.7.22

26.7.1 SCI Global Control Register 0 (SCIGCR0)

The SCIGCR0 register defines the module reset. Figure 26-8 and Table 26-4 illustrate this register.

Figure 26-8. SCI Global Control Register 0 (SCIGCR0) [offset = 00]

31	16
Reserved	
R-0	

15	1	0
Reserved		RESET
R-0		RWP-0

LEGEND: R/W = Read/Write; R = Read only; RWP = Read/Write in privileged mode only; -*n* = value after reset

Table 26-4. SCI Global Control Register 0 (SCIGCR0) Fied Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Read returns 0. Writes have no effect.
0	RESET		This bit resets the SCI module.
		0	SCI module is in reset.
		1	SCI module is out of reset.
			Note: Read/Write in privileged mode only.

26.7.2 SCI Global Control Register 1 (SCIGCR1)

The SCIGCR1 register defines the frame format, protocol, and communication mode used by the SCI. Figure 26-9 and Table 26-5 illustrate this register.

Figure 26-9. SCI Global Control Register 1 (SCIGCR1) [offset = 04h]

31	26	25	24
Reserved		TXENA	RXENA
R-0		R/W-0	R/W-0

23	18	17	16
Reserved		CONT	LOOP BACK
R-0		R/W-0	R/W-0

15	10	9	8
Reserved		POWERDOWN	SLEEP
R-0		R/WP-0	R/W-0

7	6	5	4	3	2	1	0
SW nRST	Reserved	CLOCK	STOP	PARITY	PARITY ENA	TIMING MODE	COMM MODE
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

NOTE: The SCIGCR1 Control Register Bits should not be changed during Frame Transmission or Reception.

Table 26-5. SCI Global Control Register 1 (SCIGCR1) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Read returns 0. Writes have no effect.
25	TXENA		Transmit enable. Data is transferred from SCITD to the SCITXSHF shift out register only when the TXENA bit is set.
		0	Disable transfers from SCITD to SCITXSHF.
		1	Enable SCI to transfer data from SCITD to SCITXSHF.
			Note: Data written to SCITD or the transmit multi-buffer before TXENA is set is not transmitted. If TXENA is cleared while transmission is ongoing, the data previously written to SCITD is sent.
24	RXENA		Receive enable. RXENA allows or prevents the transfer of data from SCIRXSHF to SCIRD.
		0	The receiver will not transfer data from the shift buffer to the receive buffer.
		1	The receiver will transfer data from the shift buffer to the receive buffer.
			Note: Clearing RXENA stops received characters from being transferred into the receive buffer or multi-buffers, prevents the RX status flags from being updated by receive data, and inhibits both receive and error interrupts. However, the shift register continues to assemble data regardless of the state of RXENA.
			Note: If RXENA is cleared before a frame is completely received, the data from the frame is not transferred into the receive buffer.
			Note: If RXENA is set before a frame is completely received, the data from the frame is transferred into the receive buffer. If RXENA is set while SCIRXSHF is in the process of assembling a frame, the status flags are not assured to be accurate for that frame. To ensure that the status flags correctly reflect what was detected on the bus during a particular frame, RXENA should be set before the detection of that frame.
23-18	Reserved	0	Read returns 0. Writes have no effect.
17	CONT		Continue on suspend. This bit has an effect only when a program is being debugged with an emulator, and it determines how the SCI operates when the program is suspended. The
		0	When debug mode is entered, the SCI state machine is frozen. Transmissions are halted and resume when debug mode is exited.
		1	When debug mode is entered, the SCI continues to operate until the current transmit and receive functions are complete.

Table 26-5. SCI Global Control Register 1 (SCIGCR1) Field Descriptions (continued)

Bit	Field	Value	Description
16	LOOP BACK		Loopback bit. The self-checking option for the SCI can be selected with this bit. If the SCITX and SCIRX pins are configured with SCI functionality, then the SCITX pin is internally connected to the SCIRX pin. Externally, during loop back operation, the SCITX pin outputs a high value and the SCIRX pin is in a high-impedance state. If this bit value is changed while the SCI is transmitting or receiving data, errors may result.
		0	Loop back mode is disabled.
		1	Loop back mode is enabled.
15-10	Reserved	0	Read returns 0. Writes have no effect.
9	POWERDOWN		Power down. When the POWERDOWN bit is set, the SCI attempts to enter local low-power mode. If the POWERDOWN bit is set while the receiver is actively receiving data and the wake-up interrupt is enabled, then the SCI immediately asserts an error interrupt to prevent low-power mode from being entered. Only Privilege mode writes allowed.
		0	Normal operation.
		1	Low-power mode is enabled.
8	SLEEP		SCI sleep. In a multiprocessor configuration, this bit controls the receive sleep function. Clearing this bit brings the SCI out of sleep mode.
		0	Sleep mode is disabled.
		1	Sleep mode is enabled.
			Note: The receiver still operates when the SLEEP bit is set; however, RXRDY is updated and SCIRD is loaded with new data only when an address frame is detected. The remaining receiver status flags are updated and an error interrupt is requested if the corresponding interrupt enable bit is set, regardless of the value of the SLEEP bit. In this way, if an error is detected on the receive data line while the SCI is asleep, software can promptly deal with the error condition.
			Note: The SLEEP bit is not automatically cleared when an address byte is detected.
			See Section 26.6.1 for more information on using the SLEEP bit for multiprocessor communication.
7	SWnRST		Software reset (active low).
		0	The SCI is in its reset state; no data will be transmitted or received. Writing a 0 to this bit initializes the SCI state machines and operating flags as defined in Table 26-11 and Table 26-12. All affected logic is held in the reset state until a 1 is written to this bit.
		1	The SCI is in its ready state; transmission and reception can be done. After this bit is set to 1, the configuration of the module should not change.
			Note: The SCI should only be configured while SWnRESET = 0.
6	Reserved	0	Read returns 0. Writes have no effect.
5	CLOCK		SCI internal clock enable. The CLOCK bit determines the source of the module clock on the SCICLK pin.
		0	The external SCICLK is the clock source.
		1	The internal SCICLK is the clock source.
			Note: If an external clock is selected, then the internal baud rate generator and baud rate registers are bypassed. The maximum frequency allowed for an externally sourced SCI clock is VCLK/16.
4	STOP		SCI number of stop bits per frame.
		0	One stop bit is used.
		1	Two stop bits are used.
			Note: The receiver checks for only one stop bit. However in idle-line mode, the receiver waits until the end of the second stop bit (if STOP = 1) to begin checking for an idle period.

Table 26-5. SCI Global Control Register 1 (SCIGCR1) Field Descriptions (continued)

Bit	Field	Value	Description
3	PARITY		SCI parity odd/even selection. If the PARITY ENA bit is set, PARITY designates odd or even parity.
		0	Odd parity is used.
		1	Even parity is used.
			The parity bit is calculated based on the data bits in each frame and the address bit (in address-bit mode). The start and stop fields in the frame are not included in the parity calculation.
			For odd parity, the SCI transmits and expects to receive a value in the parity bit that makes odd the total number of bits in the frame with the value of 1.
			For even parity, the SCI transmits and expects to receive a value in the parity bit that makes even the total number of bits in the frame with the value of 1.
2	PARITY ENA		Parity enable. This bit enables or disables the parity function.
		0	Parity is disabled; no parity bit is generated during transmission or is expected during reception.
		1	Parity is enabled. A parity bit is generated during transmission and is expected during reception.
1	TIMING MODE		SCI timing mode bit.
		0	Synchronous timing is used.
		1	Asynchronous timing is used.
0	COMM MODE		SCI communication mode bit.
		0	Idle-line mode is used.
		1	Address-bit mode is used.

26.7.3 SCI Set Interrupt Register (SCISETINT)

Figure 26-10 and Table 26-6 illustrate this register. SCISETINT register is used to enable the required interrupts supported by the module.

Figure 26-10. SCI Set Interrupt Register (SCISETINT) [offset = 0Ch]

31	27	26	25	24
Reserved		SET FE INT	SET OE INT	SET PE INT
R-0		R/W-0	R/W-0	R/W-0
23	19	18	17	16
Reserved		SET RX DMA ALL	SET RX DMA	SET TX DMA
R-0		R/W-0	R/W-0	R/W-0
15		10	9	8
Reserved			SET RX INT	SET TX INT
R-0			R/W-0	R/W-0
7		2	1	0
Reserved			SET WAKEUP INT	SET BRKDT INT
R-0			R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-6. SCI Set Interrupt Register (SCISETINT) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Read returns 0. Writes have no effect.
26	SET FE INT		Set framing-error interrupt. Setting this bit enables the SCI module to generate an interrupt when a framing error occurs.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read or write:* The interrupt is enabled.
25	SET OE INT		Set overrun-error interrupt. Setting this bit enables the SCI module to generate an interrupt when an overrun error occurs.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read or write:* The interrupt is enabled.
24	SET PE INT		Set parity interrupt. Setting this bit enables the SCI module to generate an interrupt when a parity error occurs.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read or write:* The interrupt is enabled.
23-19	Reserved	0	Read returns 0. Writes have no effect.
18	SET RX DMA ALL		Set receive DMA all. This bit determines if a separate interrupt is generated for the address frames sent in multiprocessor communications. When this bit is 0, RX interrupt requests are generated for address frames and DMA requests are generated for data frames. When this bit is 1, RX DMA requests are generated for both address and data frames.
		0	*Read:* The DMA request is disabled for address frames (the receive interrupt request is enabled for address frames). *Write:* No effect.
		1	*Read or write:* The DMA request is enabled for address and data frames
17	SET RX DMA		Set receiver DMA. To enable receiver DMA requests, this bit must be set. If it is cleared, interrupt requests are generated depending on bit SCISETINT.
		0	*Read:* The DMA request is disabled. *Write:* No effect.
		1	*Read or write:* The DMA request is enabled for address and data frames

Table 26-6. SCI Set Interrupt Register (SCISETINT) Field Descriptions (continued)

Bit	Field	Value	Description
16	SET TX DMA		Set transmit DMA. To enable DMA requests for the transmitter, this bit must be set. If it is cleared, interrupt requests are generated depending on SET TX INT bit (SCISETINT).
		0	*Read:* Transmit DMA request is disabled.
			Write: No effect.
		1	*Read or write:* Transmit DMA request is enabled.
15-10	Reserved	0	Read returns 0. Writes have no effect.
9	SET RX INT		Receiver interrupt enable. Setting this bit enables the SCI to generate a receive interrupt after a frame has been completely received and the data is being transferred from SCIRXSHF to SCIRD.
		0	*Read:* The interrupt is disabled.
			Write: No effect.
		1	*Read or write:* The interrupt is enabled.
8	SET TX INT		Set transmitter interrupt. Setting this bit enables the SCI to generate a transmit interrupt as data is being transferred from SCITD to SCITXSHF and the TXRDY bit is being set.
		0	*Read:* The interrupt is disabled.
			Write: No effect.
		1	*Read or write:* The interrupt is enabled.
7-2	Reserved	0	Read returns 0. Writes have no effect.
1	SET WAKEUP INT		Set wakeup interrupt. Setting this bit enables the SCI to generate a wakeup interrupt and thereby exit lowpower mode. If enabled, the wakeup interrupt is asserted when local lowpower mode is requested while the receiver is busy or if a low level is detected on the SCIRX pin during lowpower mode.
		0	*Read:* The interrupt is disabled.
			Write: No effect.
		1	*Read or write:* The interrupt is enabled.
0	SET BRKDT INT		Set breakdetect interrupt. Setting this bit enables the SCI to generate an error interrupt if a break condition is detected on the SCIRX pin.
		0	*Read:* The interrupt is disabled.
			Write: No effect.
		1	*Read or write:* The interrupt is enabled.

26.7.4 SCI Clear Interrupt Register (SCICLEARINT)

Figure 26-11 and Table 26-7 illustrate this register. SCICLEARINT register is used to clear the selected enabled interrupts with out accessing SCISETINT register.

Figure 26-11. SCI Clear Interrupt Register (SCICLEARINT) [offset = 10h]

31	27	26	25	24
Reserved		CLR FE INT	CLR OE INT	CLR PE INT
R-0		R/W-0	R/W-0	R/W-0
23	19	18	17	16
Reserved		CLR RX DMA ALL	CLR RX DMA	CLR TX DMA
R-0		R/W-0	R/W-0	R/W-0
15		10	9	8
Reserved			CLR RX INT	CLR TX INT
R-0			R/W-0	R/W-0
7		2	1	0
Reserved			CLR WAKEUP INT	CLR BRKDT INT
R-0			R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-7. SCI Clear Interrupt Register (SCICLEARINT) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Read returns 0. Writes have no effect.
26	CLR FE INT		Clear framing-error interrupt. This bit disables the framing-error interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* The interrupt is disabled.
25	CLR CE INT		Clear overrun-error interrupt. This bit disables the SCI overrun error interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* The interrupt is disabled.
24	CLR PE INT		Clear parity interrupt. This bit disables the parity error interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* The interrupt is disabled.
23-19	Reserved	0	Read returns 0. Writes have no effect.
18	CLR RX DMA ALL		Clear receive DMA all. This bit clears the receive DMA request for address frames when set. Only receive data frames generate a DMA request.
		0	*Read:* Receive DMA request for address frames is disabled; Instead, RX interrupt requests are enabled for address frames. Receive DMA requests are still enabled for data frames. *Write:* No effect.
		1	*Read:* The receive DMA request for address and data frames is enabled. *Write:* The receive DMA request for address and data frames is disabled.

Table 26-7. SCI Clear Interrupt Register (SCICLEARINT) Field Descriptions (continued)

Bit	Field	Value	Description
17	CLR RX DMA		Clear receive DMA request. This bit disables the receive DMA request when set.
		0	*Read:* The DMA request is disabled. *Write:* No effect.
		1	*Read:* The receive DMA request is enabled. *Write:* The receive DMA request for is disabled.
16	CLR TX DMA		Clear transmit DMA request. This bit disables the transmit DMA request when set.
		0	*Read:* Transmit DMA request is disabled. *Write:* No effect.
		1	*Read:* The transmit DMA request is enabled. *Write:* The transmit DMA request for is disabled.
15-10	Reserved	0	Read returns 0. Writes have no effect.
9	CLR RX INT		Clear receiver interrupt. This bit disables the receiver interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* The interrupt is disabled.
8	CLR TX INT		Clear transmitter interrupt. This bit disables the transmitter interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* The interrupt is disabled.
7-2	Reserved	0	Read returns 0. Writes have no effect.
1	CLR WAKEUP INT		Clear wakeup interrupt. This bit disables the wakeup interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* The interrupt is disabled.
0	CLR BRKDT INT		Clear breakdetect interrupt. This bit disables the break-detect interrupt when set.
		0	*Read:* The interrupt is disabled. *Write:* No effect.
		1	*Read:* The interrupt is enabled. *Write:* The interrupt is disabled.

26.7.5 SCI Set Interrupt Level Register (SCISETINTLVL)

Figure 26-12 and Table 26-8 illustrate this register. This register is used to set the interrupt level for the supported interrupts.

Figure 26-12. SCI Set Interrupt Level Register (SCISETINTLVL) [offset = 14h]

31	27	26	25	24
Reserved		SET FE INT LVL	SET OE INT LVL	SET PE INT LVL
R-0		R/W-0	R/W-0	R/W-0

23	19	18	17	16
Reserved		SET RX DMA ALL INT LVL	Reserved	
R-0		R/W-0	R-0	

15	10	9	8
Reserved		SET RX INT LVL	SET TX INT LVL
R-0		R/W-0	R/W-0

7	2	1	0
Reserved		SET WAKEUP INT LVL	SET BRKDT INT LVL
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 26-8. SCI Set Interrupt Level Register (SCISETINTLVL) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Read returns 0. Writes have no effect.
26	SET FE INT LVL	0	Set framing-error interrupt level. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The interrupt level is mapped to the INT1 line.
25	SET CE INT LVL	0	Set overrun-error interrupt level. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The interrupt level is mapped to the INT1 line.
24	SET PE INT LVL	0	Set parity error interrupt level. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The interrupt level is mapped to the INT1 line.
23-19	Reserved	0	Read returns 0. Writes have no effect.
18	SET RX DMA ALL LVL	0	Set receive DMA all interrupt levels. *Read:* The receive interrupt request for address frames is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The receive interrupt request for address frames is mapped to the INT1 line.
17-10	Reserved	0	Read returns 0. Writes have no effect.
9	SET RX INT LVL	0	Set receiver interrupt level. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The interrupt level is mapped to the INT1 line.

Table 26-8. SCI Set Interrupt Level Register (SCISETINTLVL) Field Descriptions (continued)

Bit	Field	Value	Description
8	SET TX INT LVL	0	Set transmitter interrupt level. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The interrupt level is mapped to the INT1 line.
7-2	Reserved	0	Read returns 0. Writes have no effect.
1	SET WAKEUP INT LVL	0	Set wakeup interrupt level. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The interrupt level is mapped to the INT1 line.
0	SET BRKDT INT LVL	0	Set breakdetect interrupt level. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read or write:* The interrupt level is mapped to the INT1 line.

26.7.6 SCI Clear Interrupt Level Register (SCICLEARINTLVL)

Figure 26-13 and Table 26-9 illustrate this register.

Figure 26-13. SCI Clear Interrupt Level Register (SCICLEARINTLVL) [offset = 18h]

31	27	26	25	24
Reserved		CLR FE INT LVL	CLR OE INT LVL	CLR PE INT LVL
R-0		R/W-0	R/W-0	R/W-0

23	19	18	17	16
Reserved		CLR RX DMA ALL INT LVL	Reserved	
R-0		R/W-0	R-0	

15	10	9	8
Reserved		CLR RX INT LVL	CLR TX INT LVL
R-0		R/W-0	R/W-0

7	2	1	0
Reserved		CLR WAKEUP INT LVL	CLR BRKDT INT LVL
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-9. SCI Clear Interrupt Level Register (SCICLEARINTLVL) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Read returns 0. Writes have no effect.
26	CLR FE INT LVL	0	Clear framing-error interrupt. *Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* The interrupt level is mapped to the INT0 line.

Table 26-9. SCI Clear Interrupt Level Register (SCICLEARINTLVL) Field Descriptions (continued)

Bit	Field	Value	Description
25	CLR CE INT LVL		Clear overrun-error interrupt.
		0	*Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* The interrupt level is mapped to the INT0 line.
24	CLR PE INT LVL		Clear parity interrupt.
		0	*Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* The interrupt level is mapped to the INT0 line.
23-19	Reserved	0	Read returns 0. Writes have no effect.
18	CLR RX DMA ALL LVL		Clear receive DMA interrupt level.
		0	*Read:* The receive interrupt request for address frames is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The receive interrupt request for address frames is mapped to the INT1 line. *Write:* The receive interrupt request for address frames is mapped to the INT0 line.
17-10	Reserved	0	Read returns 0. Writes have no effect.
9	CLR RX INT LVL		Clear receiver interrupt.
		0	*Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* The interrupt level is mapped to the INT0 line.
8	CLR TX INT LVL		Clear transmitter interrupt.
		0	*Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* The interrupt level is mapped to the INT0 line.
7-2	Reserved	0	Read returns 0. Writes have no effect.
1	CLR WAKEUP INT LVL		Clear wakeup interrupt.
		0	*Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* The interrupt level is mapped to the INT0 line.
0	CLR BRKDT INT LVL		Clear breakdetect interrupt.
		0	*Read:* The interrupt level is mapped to the INT0 line. *Write:* No effect.
		1	*Read:* The interrupt level is mapped to the INT1 line. *Write:* The interrupt level is mapped to the INT0 line.

26.7.7 SCI Flags Register (SCIFLR)

Figure 26-14 and Table 26-10 illustrate this register.

Figure 26-14. SCI Flags Register (SCIFLR) [offset = 1Ch]

31	27	26	25	24
Reserved		FE	OE	PE
R-0		R/W-0	R/W-0	R/W-0

23	16
Reserved	
R-0	

15	13	12	11	10	9	8
Reserved		RX WAKE	TX EMPTY	TX WAKE	RX RDY	TX RDY
R-0		R/W-0	R/W-1	R/W-0	R/W-0	R/W-1

7	4	3	2	1	0
Reserved		BUSY	IDLE	WAKE UP	BRKDT
R-0		R/W-0	R-0	R/WL-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-10. SCI Flags Register (SCIFLR) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved		Read returns 0. Writes have no effect.
26	FE		Framing error flag. This bit is effective in LIN or SCI-compatible mode. This bit is set when an expected stop bit is not found. In SCI compatibility mode, only the first stop bit is checked. The missing stop bit indicates that synchronization with the start bit has been lost and that the character is incorrectly framed. Detection of a framing error causes the SCI/LIN to generate an error interrupt if the SET FE INT bit (SCISETINT[26]). The framing error flag is cleared by the following: • Setting of the SW nRST bit • Setting of the RESET bit • A system reset • Writing a 1 to this bit • Reading the corresponding interrupt offset in SCIINTVECT0/1 • Reception of a new character/frame, depending on whether the module is in SCI compatible or LIN mode In multi-buffer mode the frame is defined in the SCIFORMAT register.
		0	*Read:* No framing error has been detected since the last clear. *Write:* No effect.
		1	*Read:* A framing error has been detected since the last clear. *Write:* The bit is cleared to 0.
25	OE		Overrun error flag. This bit is set when the transfer of data from SCIRXSHF to SCIRD overwrites unread data already in SCIRD. Detection of an overrun error causes the LIN to generate an error interrupt if the SET OE INT bit (SCISETINT[25]) is set. The OE flag is reset by the following: • Setting of the SW nRST bit • Setting of the RESET bit • A system reset • Writing a 1 to this bit • Reading the corresponding interrupt offset in SCIINTVECT0/1
		0	*Read:* No overrun error has been detected since the last clear. *Write:* No effect.
		1	*Read:* An overrun error has been detected since the last clear. *Write:* The bit is cleared to 0.

Table 26-10. SCI Flags Register (SCIFLR) Field Descriptions (continued)

Bit	Field	Value	Description
24	PE		Parity error flag. This bit is set when a parity error is detected in the received data. In SCI address-bit mode, the parity is calculated on the data and address bit fields of the received frame. In idle-line mode, only the data is used to calculate parity. An error is generated when a character is received with a mismatch between the number of 1s and its parity bit. If the parity function is disabled (SCIGCR[2] = 0), the PE flag is disabled and read as 0. Detection of a parity error causes the LIN to generate an error interrupt if the SET PE INT bit (SCISETINT[24]) is set. The PE bit is reset by the following: • Setting of the SW nRST bit • Setting of the RESET bit • A system reset • Writing a 1 to this bit • Reception of a new character or frame, depending on whether the module is in SCI compatible or LIN mode, respectively. • Reading the corresponding interrupt offset in SCIINTVECT0/1
		0	*Read:* No parity error has been detected since the last clear. *Write:* No effect.
		1	*Read:* A parity error has been detected since the last clear. *Write:* The bit is cleared to 0.
23-13	Reserved	0	Read returns 0. Writes have no effect.
12	RXWAKE		Receiver wakeup detect flag. The SCI sets this bit to indicate that the data currently in SCIRD is an address. RXWAKE is cleared by the following: • Setting of the SW nRST bit • Setting of the RESET bit • A system reset • Upon receipt of a data frame.
		0	The data in SCIRD is not an address.
		1	The data in SCIRD is an address.
11	TX EMPTY		Transmitter empty flag. This flag indicates the transmitter's buffer register(s) (SCITD/TDy) and shift register (SCITXSHF) are empty. **Note: The RESET bit, an active SW nRESET (SCIGCR1[7]), or a system reset sets this bit. This bit does not cause an interrupt request.**
		0	Transmitter buffer or shift register (or both) are loaded with data.
		1	Transmitter buffer and shift registers are both empty.
10	TXWAKE		Transmitter wakeup method select. The TXWAKE bit controls whether the data in SCITD should be sent as an address or data frame using multiprocessor communication format. This bit is set to 1 or 0 by software before a byte is written to SCITD and is cleared by the SCI when data is transferred from SCITD to SCITXSHF or by a system reset. **Note: TXWAKE is not cleared by the SW nRESET bit.** *Address-bit mode*
		0	Frame to be transmitted will be data (address bit = 0).
		1	Frame to be transmitted will be an address (address bit = 1).
			Idle-line mode
		0	The frame to be transmitted will be data.
		1	The following frame to be transmitted will be an address (writing a 1 to this bit followed by writing dummy data to the SCITD will result in a idle period of 11 bit periods before the next frame is transmitted).

Table 26-10. SCI Flags Register (SCIFLR) Field Descriptions (continued)

Bit	Field	Value	Description
9	RXRDY		Receiver ready flag. The receiver sets this bit to indicate that the SCIRD contains new data and is ready to be read by the CPU or DMA. The SCI generates a receive interrupt when RXRDY flag bit is set if the SET RX INT bit (SCISETINT[9]) is set. RXRDY is cleared by the following: • Setting of the SW nRST bit • Setting of the RESET bit • A system reset • Writing a 1 to this bit • Reading the SCIRD register in compatibility mode • Reading the last data byte RDy of the response in LIN mode **Note: The RXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0/1 register.**
		0	*Read:* No new data is in SCIRD. *Write:* No effect.
		1	*Read:* New data is ready to be read from SCIRD. *Write:* The bit is cleared to 0.
8	TXRDY		Transmitter buffer register ready flag. When set, this bit indicates that the transmit buffer is ready to get another character from a CPU or DMA write. Writing data to SCITD automatically clears this bit. This bit is set after the data of the TX buffer is shifted into the SCITXSHF register. This event can trigger a transmit interrupt after data is copied to the TX shift register SCITXSHF, if the interrupt enable bit TXINT is set. **Note: 1) TXRDY is also set to 1 by setting of the RESET bit, enabling SW nRST, or by a system reset.** **2) The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0/1 register.** **3) The transmit interrupt request can be eliminated until the next series of data written into the transmit buffers LINTD0 and LINTD1, by disabling the corresponding interrupt via the SCICLEARINT register or by disabling the transmitter via the TXENA bit (SCIGCR1[25]).**
		0	SCITD is full.
		1	SCITD is ready to receive the next character.
7-4	Reserved	0	Read returns 0. Writes have no effect.
3	BUSY		Bus busy flag. TThis bit indicates whether the receiver is in the process of receiving a frame. As soon as the receiver detects the beginning of a start bit, the BUSY bit is set to 1. When the reception of a frame is complete, the SCI clears the BUSY bit. If SET WAKEUP INT bit (SCISETINT[2]) is set and power down is requested while this bit is set, the SCI automatically prevents low-power mode from being entered and generates wakeup interrupt. The BUSY bit is controlled directly by the SCI receiver, but this bit can also be cleared by the following: • Setting the SW nRST bit • Setting of the RESET bit • A system reset occurring
		0	The receiver is not currently receiving a frame.
		1	The receiver is currently receiving a frame.
2	IDLE		SCI receiver in idle state. While this bit is set, the SCI looks for an idle period to resynchronize itself with the bit stream. The receiver does not receive any data while the bit is set. The bus must be idle for 11 bit periods to clear this bit. The SCI enters the idle state if one of the following events occurs: • A system reset • An SCI software reset • A power down • The RX pin is configured as a general I/O pin
		0	The idle period has been detected; the SCI is ready to receive.
		1	The idle period has not been detected; the SCI will not receive any data.

Table 26-10. SCI Flags Register (SCIFLR) Field Descriptions (continued)

Bit	Field	Value	Description
1	WAKEUP		Wakeup flag. This bit is set by the SCI when receiver or transmitter activity has taken the module out of power-down mode. An interrupt is generated if the SET WAKEUP INT bit (SCISETINT[2]) is set. It is cleared by the following: • Setting of the SW nRST bit • Setting of the RESET bit • A system reset • Writing a 1 to this bit • Reading the corresponding interrupt offset in SCIINTVECT0/1 For compatibility mode, see the SCI document for more information on low-power mode.
		0	*Read:* The module will not wake up from power-down mode. *Write:* No effect.
		1	*Read:* Wake up from power-down mode. *Write:* The bit is cleared to 0.
0	BRKDT		SCI break-detect flag. This bit is set when the SCI detects a break condition on the LINRX pin. A break condition occurs when the SCIRX pin remains continuously low for at least 10 bits after a missing first stop bit, that is, after a framing error. Detection of a break condition causes the SCI to generate an error interrupt if the SET BRKDT INT bit (SCISETINT[0]) is set. The BRKDT bit is reset by the following: • Setting of the SW nRST bit • Setting of the RESET bit • A system reset • Writing a 1 to this bit • Reading the corresponding interrupt offset in SCIINTVECT0/1
		0	*Read:* No break condition has been detected since the last clear. *Write:* No effect.
		1	*Read:* A break condition has been detected. *Write:* The bit is cleared to 0.

Table 26-11. SCI Receiver Status Flags

SCI Flag	Register	Bit	Value After SW nRESET (1)
FE	SCIFLR	26	0
OE	SCIFLR	25	0
PE	SCIFLR	24	0
RXWAKE	SCIFLR	12	0
RXRDY	SCIFLR	9	0
BRKDT	SCIFLR	0	0

(1) The flags are frozen with their reset value while SW nRESET = 0.

Table 26-12. SCI Transmitter Status Flags

SCI Flag	Register	Bit	Value After SW nRESET (1)
TX EMPTY	SCIFLR	11	1
TXRDY	SCIFLR	8	1

(1) The flags are frozen with their reset value while SW nRESET = 0.

26.7.8 SCI Interrupt Vector Offset 0 (SCIINTVECT0)

Figure 26-15 and Table 26-13 illustrate this register.

Figure 26-15. SCI Interrupt Vector Offset 0 (SCIINTVECT0) [offset = 20h]

31	16
Reserved	
R-0	

15	4	3	0
Reserved		INTVECT0	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-13. SCI Interrupt Vector Offset 0 (SCIINTVECT0) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Read returns 0. Writes have no effect.
3-0	INVECT0	0-Fh	Interrupt vector offset for INT0. This register indicates the offset for interrupt line INT0. A read to this register updates its value to the next highest priority pending interrupt in SCIFLR and clears the flag in SCIFLR corresponding to the offset that was read. See Table 26-1 for a list of the interrupts. **Note: The flags for the receive (SCIFLR[9]) and the transmit (SCIFLR[8]) interrupt cannot be cleared by reading the corresponding offset vector in this register (see detailed description in SCIFLR register).**

26.7.9 SCI Interrupt Vector Offset 1 (SCIINTVECT1)

Figure 26-16 and Table 26-14 illustrate this register.

Figure 26-16. SCI Interrupt Vector Offset 1 (SCIINTVECT1) [offset = 24h]

31	16
Reserved	
R-0	

15	4	3	0
Reserved		INTVECT1	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

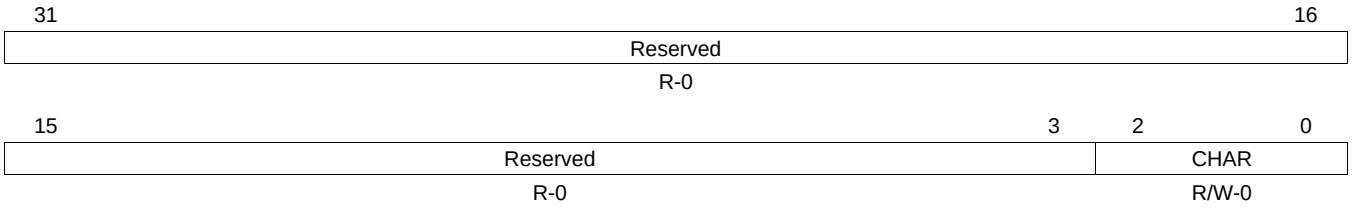
Table 26-14. SCI Interrupt Vector Offset 1 (SCIINTVECT1) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Read returns 0. Writes have no effect.
3-0	INVECT1	0-Fh	Interrupt vector offset for INT1. This register indicates the offset for interrupt line INT1. A read to this register updates its value to the next highest priority pending interrupt in SCIFLR and clears the flag in SCIFLR corresponding to the offset that was read. See Table 26-1 for list of interrupts. **Note: The flags for the receive (SCIFLR[9]) and the transmit (SCIFLR[8]) interrupt cannot be cleared by reading the corresponding offset vector in this register (see detailed description in SCIFLR register).**

26.7.10 SCI Format Control Register (SCIFORMAT)

Figure 26-17 and Table 26-15 illustrate this register.

Figure 26-17. SCI Format Control Register (SCIFORMAT) [offset = 28h]



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-15. SCI Format Control Register (SCIFORMAT) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2-0	CHAR		Character length control bits. These bits set the SCI character length from 1 to 8 bits. **When data of fewer than eight bits in length is received, it is left-justified in SCIRD and padded with trailing zeros.** **Data read from the SCIRD should be shifted by software to make the received data right-justified.** **Data written to the SCITD should be right-justified but does not need to be padded with leading zeros.**
		0	The character is 1 bit long.
		1h	The character is 2 bits long.
		2h	The character is 3 bits long.
		3h	The character is 4 bits long.
		4h	The character is 5 bits long.
		5h	The character is 6 bits long.
		6h	The character is 7 bits long.
		7h	The character is 8 bits long.

26.7.11 Baud Rate Selection Register (BRS)

This section describes the baud rate selection register. Figure 26-18 and Table 26-16 illustrate this register.

Figure 26-18. Baud Rate Selection Register (BRS) [offset = 2Ch]

31	24	23	16
Reserved		BAUD	
R-0		R/W-0	

15	0
BAUD	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 26-16. Baud Rate Selection Register (BRS) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Read returns 0. Writes have no effect.
23-0	BAUD	0-FF FFFFh	SCI 24-bit baud selection. The SCI has an internally generated serial clock determined by the VCLK and the prescalers BAUD in this register. The SCI uses the 24-bit integer prescaler BAUD value of this register to select one of over 16,700,000. The baud rate can be calculated using the following formulas: $\text{Asynchronous baud value} = \left(\frac{\text{VBUSPLCK Frequency}}{16(\text{Baud} + 1)}\right)$ (17) $\text{Isosynchronous baud value} = \left(\frac{\text{VBUSPLCK Frequency}}{\text{Baud} + 1}\right)$ (18) For BAUD = 0, $\text{Asynchronous baud value} = \left(\frac{\text{VCLK Frequency}}{32}\right)$ (19) $\text{Isosynchronous baud value} = \left(\frac{\text{VCLK Frequency}}{2}\right)$ (20) Table 26-17 contains comparative baud values for different P values, with VCLK = 50 MHz, for asynchronous mode..

Table 26-17. Comparative Baud Values for Different P Values, Asynchronous Mode [(1)(2)]

24-Bit Register Value		Baud Selected		Percent Error
Decimal	Hex	Ideal	Actual	
26	00001A	115200	115740	0.47
53	000035	57600	57870	0.47
80	000050	38400	38580	0.47
162	0000A2	19200	19172	-0.15
299	00012B	10400	10417	0.16
325	000145	9600	9586	-0.15
399	00018F	7812.5	7812.5	0.00
650	00028A	4800	4800	0.00
15624	003BA0	200	200	0.00
624999	098967	5	5	0.00

(1) VCLK = 50 MHz
(2) Values are in decimal except for column 2.

26.7.12 SCI Data Buffers (SCIED, SCIRD, SCITD)

The SCI has three addressable registers in which transmit and receive data is stored.

26.7.12.1 Receiver Emulation Data Buffer (SCIED)

The SCIED register is addressed at a location different from SCIRD, but is physically the same register. Figure 26-19 and Table 26-18 illustrate this register.

Figure 26-19. Receiver Emulation Data Buffer (SCIED) [offset = 30h]

31								16
Reserved								
R-0								
15				8	7			0
Reserved					ED			
R-0					R-0			

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-18. Receiver Emulation Data Buffer (SCIED) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Read returns 0. Writes have no effect.
7-0	ED	0-FFh	Emulator data. Reading SCIED[7:0] does not clear the RXRDY flag (SCIFLR[9]), unlike reading SCIRD. This register should be used only by an emulator that must continually read the data buffer without affecting the RXRDY flag.

26.7.12.2 Receiver Data Buffer (SCIRD)

This register provides a location for the receiver data. Figure 26-20 and Table 26-19 illustrate this register.

Figure 26-20. Receiver Data Buffer (SCIRD) [offset = 34h]

31								16
Reserved								
R-0								
15				8	7			0
Reserved					RD			
R-0					R-0			

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-19. Receiver Data Buffer (SCIRD) Field Descriptions

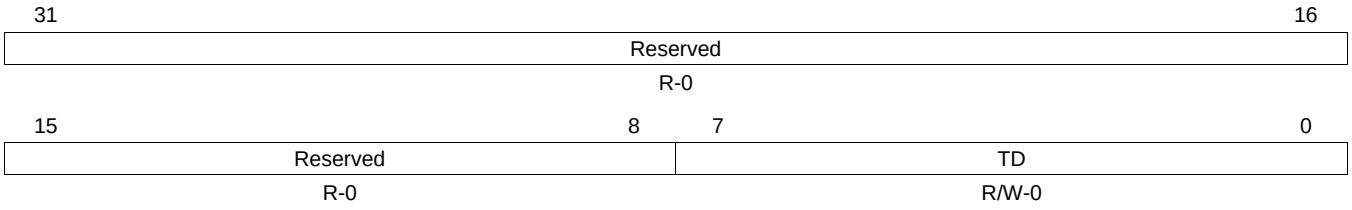
Bit	Field	Value	Description
31-8	Reserved	0	Read returns 0. Writes have no effect.
7-0	RD	0-FFh	Receiver data. When a frame has been completely received, the data in the frame is transferred from the receiver shift register SCIRXSHF to this register. As this transfer occurs, the RXRDY flag (SCIFLR[9]) is set and a receive interrupt is generated if SET RX INT bit (SCISETINT[9]) is set. **Note: When the data is read from SCIRD, the RXRDY flag (SCIFLR[9]) is automatically cleared.**

NOTE: When the SCI receives data that is fewer than eight bits in length, it loads the data into this register in a left-justified format padded with trailing zeros. Therefore, the user software should perform a logical shift on the data by the correct number of positions to make it right justified.

26.7.12.3 Transmit Data Buffer Register (SCITD)

Data to be transmitted is written to the SCITD register. Figure 26-21 and Table 26-20 illustrate this register.

Figure 26-21. Transmit Data Buffer Register (SCITD) [offset = 38h]



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-20. Transmit Data Buffer Register (SCITD) Field Descriptions

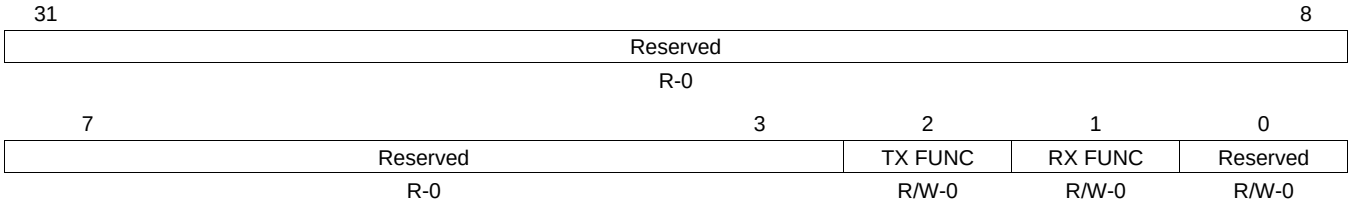
Bit	Field	Value	Description
31-8	Reserved	0	Read returns 0. Writes have no effect.
7-0	TD	0-FFh	Transmit data. Data to be transmitted is written to the SCITD register. The transfer of data from this register to the transmit shift register SCITXSHF sets the TXRDY flag (SCIFLR[8]), which indicates that SCITD is ready to be loaded with another byte of data. **Note: If SET TX INT bit (SCISETINT[8] is set, this data transfer also causes an interrupt.**

NOTE: Data written to the SCITD register that is fewer than eight bits long must be right-justified, but it does not need to be padded with leading zeros.

26.7.13 SCI Pin I/O Control Register 0 (SCIPIO0)

Figure 26-22 and Table 26-21 illustrate this register.

Figure 26-22. SCI Pin I/O Control Register 0 (SCIPIO0) [offset = 3Ch]



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-21. SCI Pin I/O Control Register 0 (SCIPIO0) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX FUNC		Transfer function. This bit defines the function of pin SCITX.
		0	SCITX is a general-purpose digital I/O pin.
		1	SCITX is the SCI transmit pin.
1	RX FUNC		Receive function. This bit defines the function of pin SCIRX.
		0	SCIRX is a general-purpose digital I/O pin.
		1	SCIRX is the SCI receive pin.
0	Reserved	0	Writes have no effect.

26.7.14 SCI Pin I/O Control Register 1 (SCIPIO1)

Figure 26-23 and Table 26-22 illustrate this register.

Figure 26-23. SCI Pin I/O Control Register 1 (SCIPIO1) [offset = 40h]

31				8
Reserved				
R-0				
7	**3**	**2**	**1**	**0**
Reserved		TX DIR	RX DIR	Reserved
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 26-22. SCI Pin I/O Control Register 1 (SCIPIO1) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX DIR		Transmit pin direction. This bit determines the data direction on the SCITX pin if it is configured with general-purpose I/O functionality (TX FUNC = 0). See Table 26-23 for the SCITX pin control with this bit and others.
		0	SCITX is a general-purpose input pin.
		1	SCITX is a general-purpose output pin.
1	RX DIR		Receive pin direction. This bit determines the data direction on the SCIRX pin if it is configured with general-purpose I/O functionality (RX FUNC = 0). See Table 26-24 for the SCIRX pin control with this bit and others.
		0	SCIRX is a general-purpose input pin.
		1	SCIRX is a general-purpose output pin.
0	Reserved	0	Writes have no effect.

Table 26-23. SCITX Pin Control

Function	TX IN [1]	TX OUT	TX FUNC	TX DIR
SCITX	X	X	1	X
General-purpose input	X	X	0	0
General-purpose output, high	X	1	0	1
General-purpose output, low	X	0	0	1

(1) TX IN is a read-only bit. Its value always reflects the level of the SCITX pin.

Table 26-24. SCIRX Pin Control

Function	RX IN [1]	RX OUT	RX FUNC	RX DIR
SCIRX	X	X	1	X
General-purpose input	X	X	0	0
General-purpose output, high	X	1	0	1
General-purpose output, low	X	0	0	1

(1) RX IN is a read-only bit. Its value always reflects the level of the SCIRX pin.

26.7.15 SCI Pin I/O Control Register 2 (SCIPIO2)

Figure 26-24 and Table 26-25 illustrate this register.

Figure 26-24. SCI Pin I/O Control Register 2 (SCIPIO2) [offset = 44h]

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		TX IN	RX IN	Reserved
R-0		R-X	R-X	R-X

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset; *-x* = Indeterminate

Table 26-25. SCI Pin I/O Control Register 2 (SCIPIO2) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX IN		Transmit pin in. This bit contains the current value on the SCITX pin.
		0	The SCITX pin is at logic low (0).
		1	The SCITX pin is at logic high (1).
1	RX IN		Receive pin in. This bit contains the current value on the SCIRX pin.
		0	The SCIRX pin is at logic low (0).
		1	The SCIRX pin is at logic high (1).
0	Reserved	0	Writes have no effect.

26.7.16 SCI Pin I/O Control Register 3 (SCIPIO3)

Figure 26-25 and Table 26-26 illustrate this register.

Figure 26-25. SCI Pin I/O Control Register 3 (SCIPIO3) [offset = 48h]

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		TX OUT	RX OUT	Reserved
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-26. SCI Pin I/O Control Register 3 (SCIPIO3) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX OUT		Transmit pin out. This pin specifies the logic to be output on pin SCITX if the following conditions are met: • TX FUNC = 0 (SCITX pin is a general-purpose I/O.) • TX DIR = 1 (SCITX pin is a general-purpose output.) See Table 26-23 for an explanation of this bit's effect in combination with other bits.
		0	The output on the SCITX is at logic low (0).
		1	The output on the SCITX pin is at logic high (1). (Output voltage is V_{OH} or higher if TXPDR = 0 and output is in high impedance state if TXPDR = 1)
1	RX OUT		Receive pin out. This bit specifies the logic to be output on pin SCIRX if the following conditions are met: • RX FUNC = 0 (SCIRX pin is a general-purpose I/O.) • RX DIR = 1 (SCIRX pin is a general-purpose output.) See Table 26-24 for an explanation of this bit's effect in combination with the other bits.
		0	The output on the SCIRX pin is at logic low (0).
		1	The output on the SCIRX pin is at logic high (1). (Output voltage is V_{OH} or higher if RXPDR = 0, and output is in high impedance state if RXPDR = 1)
0	Reserved	0	Writes have no effect.

26.7.17 SCI Pin I/O Control Register 4 (SCIPIO4)

Figure 26-26 and Table 26-27 illustrate this register.

Figure 26-26. SCI Pin I/O Control Register 4 (SCIPIO4) [offset = 4Ch]

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		TX SET	RX SET	Reserved
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-27. SCI Pin I/O Control Register 4 (SCIPIO4) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX SET		Transmit pin set. This bit sets the logic to be output on pin SCITX if the following conditions are met: • TX FUNC = 0 (SCITX pin is a general-purpose I/O.) • TX DIR = 1 (SCITX pin is a general-purpose output.) See Table 26-23 for an explanation of this bit's effect in combination with other bits.
		0	*Read:* The output on SCITX is at logic low (0). *Write:* No effect.
		1	*Read or write:* The output on SCITX is at logic high (1).
1	RX SET		Receive pin set. This bit sets the data to be output on pin SCIRX if the following conditions are met: • RX FUNC = 0 (SCIRX pin is a general-purpose I/O.) • RX DIR = 1 (SCIRX pin is a general-purpose output.) See Table 26-24 for an explanation of this bit's effect in combination with the other bits.
		0	*Read:* The output on SCIRX is at logic low (0). *Write:* No effect.
		1	*Read or write:* The output on SCIRX is at logic high (1).
0	Reserved	0	Writes have no effect.

26.7.18 SCI Pin I/O Control Register 5 (SCIPIO5)

Figure 26-27 and Table 26-28 illustrate this register.

Figure 26-27. SCI Pin I/O Control Register 5 (SCIPIO5) [offset = 50h]

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		TX CLR	RX CLR	Reserved
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-28. SCI Pin I/O Control Register 5 (SCIPIO5) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX CLR		Transmit pin clear. This bit clears the logic to be output on pin SCITX if the following conditions are met: • TX FUNC = 0 (SCITX pin is a general-purpose I/O.) • TX DIR = 1 (SCITX pin is a general-purpose output.)
		0	*Read:* The output on SCITX is at logic low (0). *Write:* No effect.
		1	*Read:* The output on SCITX is at logic high (1). *Write:* The output on SCITX is at logic low (0).
1	RX CLR		Receive pin clear. This bit clears the logic to be output on pin SCIRX if the following conditions are met: • RX FUNC = 0 (SCIRX pin is a general-purpose I/O.) • RX DIR = 1 (SCIRX pin is a general-purpose output.)
		0	*Read:* The output on SCIRX is at logic low (0). *Write:* No effect.
		1	*Read:* The output on SCIRX is at logic high (1). *Write:* The output on SCIRX is at logic low (0).
0	Reserved	0	Writes have no effect.

26.7.19 SCI Pin I/O Control Register 6 (SCIPIO6)

Figure 26-28 and Table 26-29 illustrate this register.

Figure 26-28. SCI Pin I/O Control Register 6 (SCIPIO6) [offset = 54h]

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		TX PDR	RX PDR	Reserved
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26-29. SCI Pin I/O Control Register 6 (SCIPIO6) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX PDR		Transmit pin open drain enable. This bit enables open-drain capability in the output pin SCITX if the following conditions are met: • TX FUNC = 0 (SCITX pin is a general-purpose I/O.) • TX DIR = 1 (SCITX pin is a general-purpose output.)
		0	Open drain functionality is disabled; the output voltage is V_{OL} or lower if TXOUT = 0 and V_{OH} or higher if TXOUT = 1.
		1	Open drain functionality is enabled; the output voltage is V_{OL} or lower if TXOUT = 0 and high impedance if TXOUT = 1.
1	RX PDR		Receive pin open drain enable. This bit enables open-drain capability in the output pin SCIRX if the following conditions are met: • RX FUNC = 0 (SCIRX pin is a general-purpose I/O.) • RX DIR = 1 (SCIRX pin is a general-purpose output.)
		0	Open drain functionality is disabled; the output voltage is V_{OL} or lower if RXOUT = 0 and V_{OH} or higher if RXOUT = 1.
		1	Open drain functionality is enabled; the output voltage is V_{OL} or lower if RXOUT = 0 and high impedance if RXOUT = 1.
0	Reserved	0	Writes have no effect.

26.7.20 SCI Pin I/O Control Register 7 (SCIPIO7)

Figure 26-29 and Table 26-30 illustrate this register.

Figure 26-29. SCI Pin I/O Control Register 7 (SCIPIO7) [offset = 58h]

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		TX PD	RX PD	Reserved
R-0		R/W-n	R/W-n	R/W-n

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset, Refer to the Terminal Functions in the device datasheet for default pin settings.

Table 26-30. SCI Pin I/O Control Register 7 (SCIPIO7) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX PD		Transmit pin pull control disable. This bit disables pull control capability on the input pin SCITX.
		0	The pull control on the SCITX pin is enabled.
		1	The pull control on the SCITX pin is disabled.
1	RX PD		Receive pin pull control disable. This bit disables pull control capability on the input pin SCIRX.
		0	Pull control on the SCIRX pin is enabled.
		1	Pull control on the SCIRX pin is disabled.
0	Reserved	0	Writes have no effect.

26.7.21 SCI Pin I/O Control Register 8 (SCIPIO8)

Figure 26-30 and Table 26-31 illustrate this register.

Figure 26-30. SCI Pin I/O Control Register 8 (SCIPIO8) [offset = 5Ch]

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		TX PSL	RX PSL	Reserved
R-0		R/W-n	R/W-n	R/W-n

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset, Refer to the Terminal Functions in the device datasheet for default pin settings.

Table 26-31. SCI Pin I/O Control Register 8 (SCIPIO8) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Read returns 0. Writes have no effect.
2	TX PSL		TX pin pull select. This bit selects pull type in the input pin SCITX.
		0	The SCITX pin is a pull down.
		1	The SCITX pin is a pull up.
1	RX PSL		RX pin pull select. This bit selects pull type in the input pin SCIRX.
		0	The SCIRX pin is a pull down.
		1	The SCIRX pin is a pull up.
0	Reserved	0	Writes have no effect.

26.7.22 Input/Output Error Enable (IODFTCTRL) Register

Figure 26-31 and Table 26-32 illustrate this register. After the basic SCI module configuration, enable the required Error mode to be created followed by IODFT Key enable.

NOTE:

1. All the bits are used in IODFT mode only.
2. Each IODFT are expected to be checked individually.

Figure 26-31. Input/Output Error Enable Register (IODFTCTRL) [offset = 90h]

31	27	26	25	24
Reserved		FEN	PEN	BRKDTENA
R-0		R/W-0	R/W-0	R/W-0

23	21	20	19	18	16
Reserved		PIN SAMPLE MASK		TX SHIFT	
R-0		R/W-0		R/W-0	

15	12	11	8
Reserved		IODFTENA	
R-0		R/WP-0	R/WP-1

7	2	1	0
Reserved		LPB ENA	RXPENA
R-0		R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -*n* = value after reset

Table 26-32. Input/Output Error Enable Register (IODFTCTRL) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Read returns 0. Writes have no effect.
26	FEN		Frame error enable. This bit is used to create a frame error.
		0	No error is created.
		1	The stop bit received is ANDed with 0 and passed to the stop bit check circuitry.
25	PEN		Parity error enable. This bit is used to create a parity error.
		0	No parity error occurs.
		1	The parity bit received is toggled so that a parity error occurs.
24	BRKD TENA		Break detect error enable. This bit is used to create a BRKDT error.
		0	No error is created.
		1	The stop bit of the frame is ANDed with 0 and passed to the RSM so that a frame error occurs. Then the RX pin is forced to continuous low for 10 T_{BITS} so that a BRKDT error occurs.
32-21	Reserved	0	Read returns 0. Writes have no effect.
20-19	PIN SAMPLE MASK		Pin sample mask. These bits define the sample number at which the TX pin value that is being transmitted will be inverted to verify the receive pin samples majority detection circuitry.
		0	No mask is used.
		1h	Invert the TX Pin value at 7th SCLK.
		2h	Invert the TX Pin value at 8th SCLK.
		3h	Invert the TX Pin value at 9th SCLK.

Table 26-32. Input/Output Error Enable Register (IODFTCTRL) Field Descriptions (continued)

Bit	Field	Value	Description
18-16	TX SHIFT		Transmit shift. These bits define the amount by which the value on TX pin is delayed so that the value on the RX pin is asynchronous. This feature is not applicable to the start bit.
		0	No delay occurs.
		1h	The value is delayed by 1 SCLK.
		2h	The value is delayed by 2 SCLK.
		3h	The value is delayed by 3 SCLK.
		4h	The value is delayed by 4 SCLK.
		5h	The value is delayed by 5 SCLK.
		6h	The value is delayed by 6 SCLK.
		7h	No delay occurs.
15-12	Reserved	0	Read returns 0. Writes have no effect.
11-8	IODFTENA		IODFT enable key. Write access permitted in Privilege mode only.
		Ah	IODFT is enabled.
		All Others	IODFT is disabled.
7-2	Reserved	0	Read returns 0. Writes have no effect.
1	LPBENA		Module loopback enable. Write access permitted in Privilege mode only.
			Note: In analog loopback mode the complete communication path through the I/Os can be tested, whereas in digital loopback mode the I/O buffers are excluded from this path.
		0	Digital loopback is enabled.
		1	Analog loopback is enabled in module I/O DFT mode when IODFTENA = 1010.
0	RXPENA		Module analog loopback through receive pin enable. Write access permitted in Privilege mode only. This bit defines whether the I/O buffers for the transmit or the receive pin are included in the communication path (in analog loopback mode)
		0	Analog loopback through the transmit pin is enabled.
		1	Analog loopback through the receive pin is enabled.

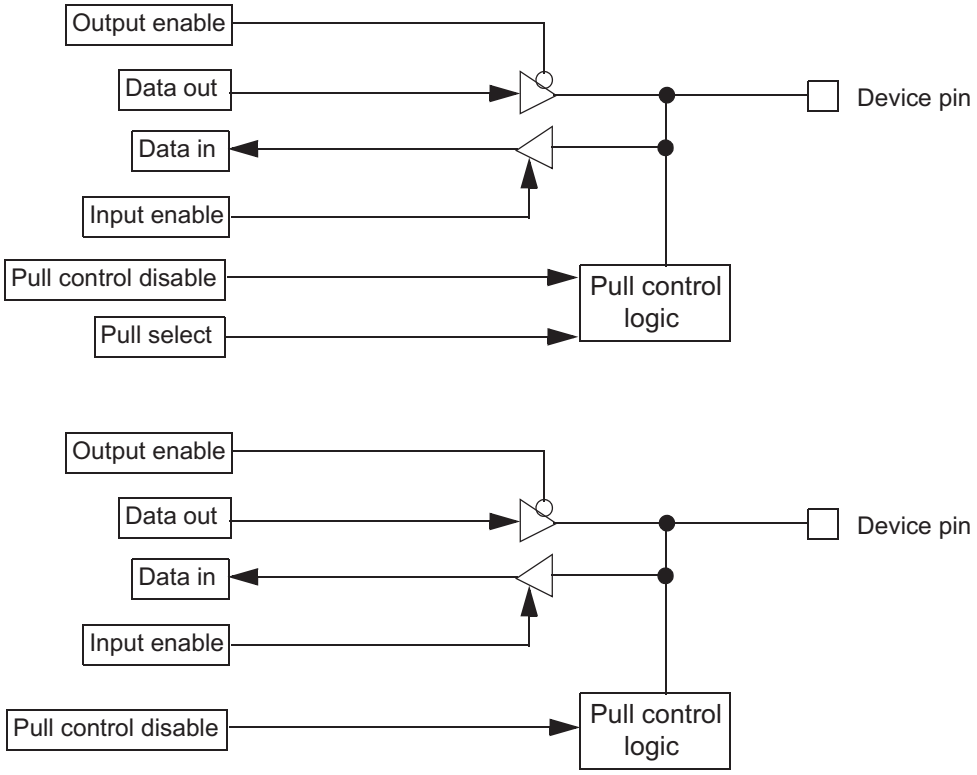
26.8 GPIO Functionality

The following sections apply to all device pins that can be configured as functional or general-purpose I/O pins.

26.8.1 GPIO Functionality

Figure 26-32 illustrates the GPIO functionality.

Figure 26-32. GPIO Functionality



26.8.2 Under Reset

The following apply if a device is under reset:

- Pull control. The reset pull control on the pins is enabled or disabled depending on a device-specific option. This feature is configurable for each module separately.
- Input buffer. The input buffer is enabled.
- Output buffer. The output buffer is disabled.

26.8.3 Out of Reset

The following apply if the device is out of reset:

- Pull control. The pull control is enabled by clearing the PD (pull control disable) bit in the SCIPIO7 register (Section 26.7.20). In this case, if the PSL (pull select) bit in the SCIPIO8 register (Section 26.7.21) is set, the pin will have a pull-up. If the PSL bit is cleared, the pin will have a pull-down. If the PD bit is set in the control register, there is no pull-up or pull-down on the pin.
- Input buffer. The input buffer is always enabled in functional mode.

NOTE: The pull-disable logic depends on the pin direction. It is independent of whether the device is in I/O or functional mode. If the pin is configured as output or transmit, then the pulls are disabled automatically. If the pin is configured as input or receive, the pulls are enabled or disabled depending on bit PD in the pull disable register SCIPIO7 (Section 26.7.20).

- Output buffer. A pin can be driven as an output pin if the TX DIR bit is set in the pin direction control register (SCIPIO1; Section 26.7.14) AND the open-drain feature is not enabled in the SCIPIO6 register (Section 26.7.19).

26.8.4 Open-Drain Feature Enabled on a Pin

The following apply if the open-drain feature is enabled on a pin:

- The output buffer is enabled if a low signal is being driven on to the pin.
- The output buffer is disabled (the direction control signal DIR is internally forced low) if a high signal is being driven on to the pin.

NOTE: The open-drain feature is available only in I/O mode (SCIPIO0; Section 26.7.13).

26.8.5 Summary

The behavior of the input buffer, output buffer, and the pull control is summarized in Table 26-33.

Table 26-33. Input Buffer, Output Buffer, and Pull Control Behavior as GPIO Pins

Device under Reset?	Pin Direction (DIR) [1][2]	Pull Disable (PULDIS) [1][3]	Pull Select (PULSEL) [1][4]	Pull Control	Output Buffer	Input Buffer
Yes	X	X	X	Device- and module-specific	Disabled	Enabled
No	0	0	0	Pull down	Disabled	Enabled
No	0	0	1	Pull up	Disabled	Enabled
No	0	1	0	Disabled	Disabled	Enabled
No	0	1	1	Disabled	Disabled	Enabled
No	1	X	X	Disabled	Enabled	Enabled

(1) X = Don't care
(2) DIR = 0 for input, 1 for output
(3) PULDIS = 0 for enabling pull control
= 1 for disabling pull control
(4) PULSEL= 0 for pull-down functionality
= 1 for pull-up functionality

Debug Architecture

The debug subsystem contains OneMCUDebugss at its core that allows the JTAG interface access to device components. The debug subsystem is designed to provide the following debug features:

- JTAG debug access to debug resources, mapped through an ARM SWJ-DP and TI ICEPickM scan module
- System memory access without halting the processor
- Trace for C674x DSP
- ETM-based trace for ARM R4F
- Cross trigger to halt and restart MSS and DSP, based on events such as watchdog, timers, DMA, and time-stamp events
- Capability to read the device ID

27.1 Debug Subsystem

Figure 27-1 shows the top-level block diagram of the debug subsystem. Debug acts as master controller to the rest of the debug-enabled subsystems on the device. Trace data is collected from the MSS CR4 and DSP core, and can either be stored in the emulate trace buffer (ETB) or sent over the device trace pins. Cross triggering is established using cross trigger interface (CTI). The debug system CTI interface is connected to the ARM Cortex R4F and DSP core to allow cross triggering of capture based on various events within the device.

Figure 27-1. Block Diagram of the 14xx Debug Subsystem Interconnect

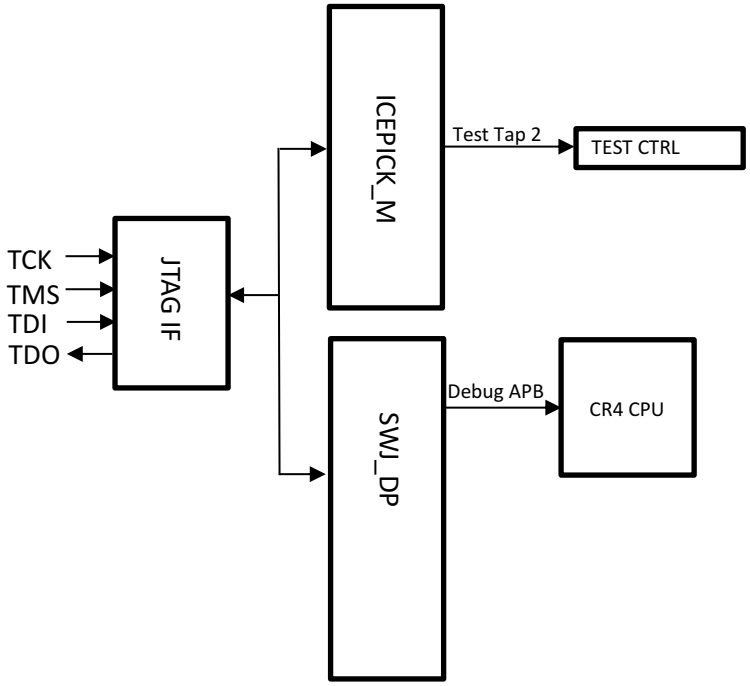


Table 27-1 shows the address map for the debug subsystem.

Table 27-1. 14xx Debug Subsystem Address Map

APB PORT	Block Name	Start Address	End Address
APB MASTER 0	MSS APB	0xFFA10000	0xFFA1FFFF
APB MASTER 1	Reserved	0xFFA20000	0xFFA2FFFF

Figure 27-2. Block Diagram of the 16xx/18xx/68xx Debug Subsystem Interconnect

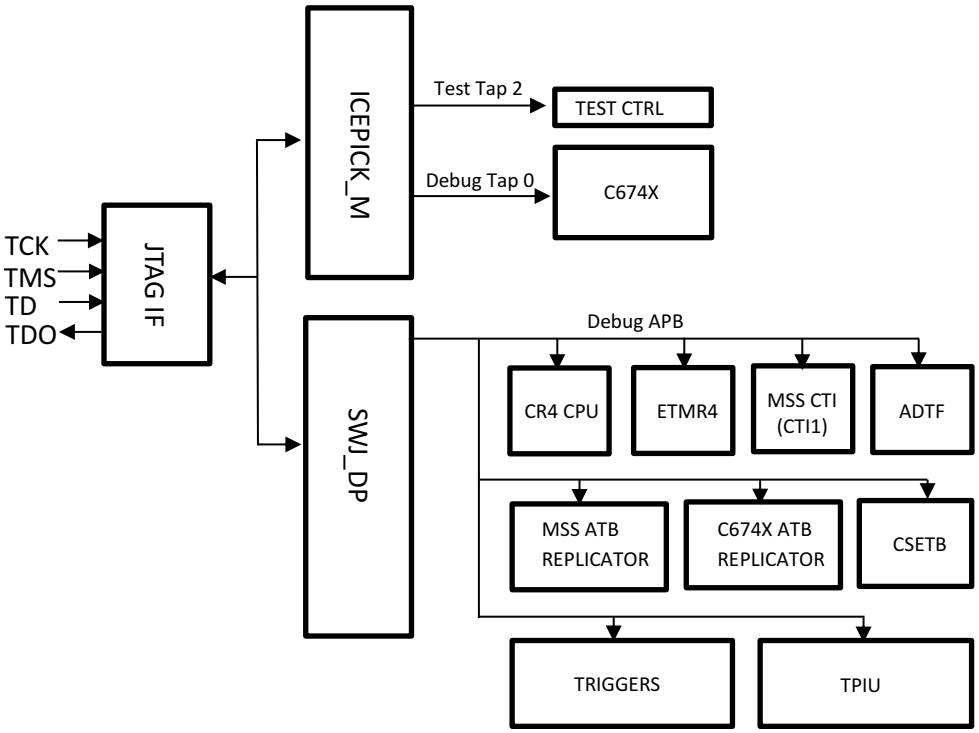


Table 27-2. 16xx/18xx/68xx Debug Subsystem Address Map

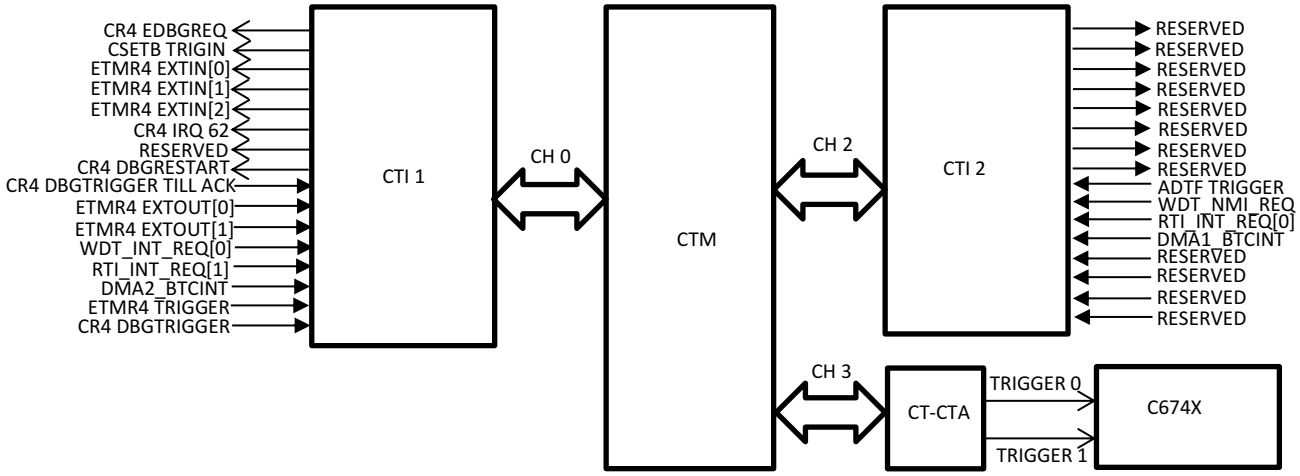
APB PORT	Block Name	Start Address	End Address
APB MASTER 0	MSS CTI 2 (CTI2)	0xFFA01000	0xFFA01FFF
APB MASTER 0	TPIU	0xFFA02000	0xFFA02FFF
APB MASTER 1	MSS APB	0xFFA10000	0xFFA17FFF
APB MASTER 1	MSS ETMR4	0xFFA18000	0xFFA1BFFF
APB MASTER 1	MSS CTI (CTI1)	0xFFA1C000	0xFFA1FFFF
APB MASTER 2	Reserved	0xFFA20000	0xFFA27FFF
APB MASTER 3	GEM ADTF	0xFFA30000	0xFFA33FFF
APB MASTER 3	GEM ATB REPLICATOR	0xFFA34000	0xFFA37FFF
APB MASTER 3	MSS ATB REPLICATOR	0xFFA38000	0xFFA3BFFF
APB MASTER 3	CSETB	0xFFA3C000	0xFFA3FFFF

27.2 Triggering Subsystem

The trigger subsystem consists of CTI modules connected by the CTI channel interface. A trigger connected to any of the CTI can be routed on the trigger of the other CTI. Triggers from MSS CR4 and ETMR4 are connected to MSS CTI. Triggers from MSS CTI are connected to MSS CR4, ETMR4, and CSETB. MSS CTI also received triggers from DMA2, RTI, and Watchdog.

The second CTI module is present inside the debug subsystem. This CTI receives triggers from ADTF, DMA, RTI, and WDT. GEM. The C674x DSP can be halted by any trigger in the subsystem. When GEM trace is enabled, triggers are generated by ADTF by detecting the trigger packets in the trace stream.

Figure 27-3. 16xx/18xx/68xx Cross Trigger Subsystem



27.2.1 Cross Trigger Input Tables

Table 27-3. MSS CTI Trigger Inputs

MSS CTI Trigger Number	MSS CTI Trigger Input
0	MSS CR4 DBGTRIGGER held active till acknowledged
1	MSS ETMR4 EXTOUT[0]
2	MSS ETMR4 EXTOUT[1]
3	wdt_int_req[0]
4	rti_int_req[1]
5	dma2_btcint
6	ETMR4 Trigger
7	MSS CR4 DBGTRIGGER

Table 27-4. MSS CTI Trigger Outputs

MSS CTI Trigger Number	MSS CTI Trigger Output
0	MSS CR4 EDBGRQ
1	CSETB Trigin
2	MSS ETMR4 EXTIN[0]
3	MSS ETMR4 EXTIN[1]
4	MSS ETMR4 EXTIN[2]
5	MSS CR4 IRQ 62
6	Open
7	MSS CR4 DBGRESTART

Table 27-5. Debug SS CTI Trigger Inputs

OneMCU Debugss CTI Trigger number	DebugSS CTI Trigger input
0	ADTF Trigger
1	wdt_nmi_req
2	rti_int_req[0]
3	dma1_btcint

Safety

Topic **Page**

28.1 Dual Clock Comparator (DCC) 3286
28.2 Error Signaling Module (ESM) 3303
28.3 Cyclic Redundancy Check (CRC) 3340
28.4 Programmable Built-In Self-Test (PBIST) 3425
28.5 Self-Test Controller (STC) 3441
28.6 Core Clock Comparator (CCC) Module 3526

28.1 Dual Clock Comparator (DCC)

This section describes the dual-clock comparator (DCC) module.

28.1.1 Introduction

The primary purpose of a DCC module is to measure the frequency of a clock signal using a second known clock signal as a reference. This capability can be used to ensure the correct frequency range for several different device clock sources, thereby enhancing the system safety metrics.

28.1.1.1 Main Features

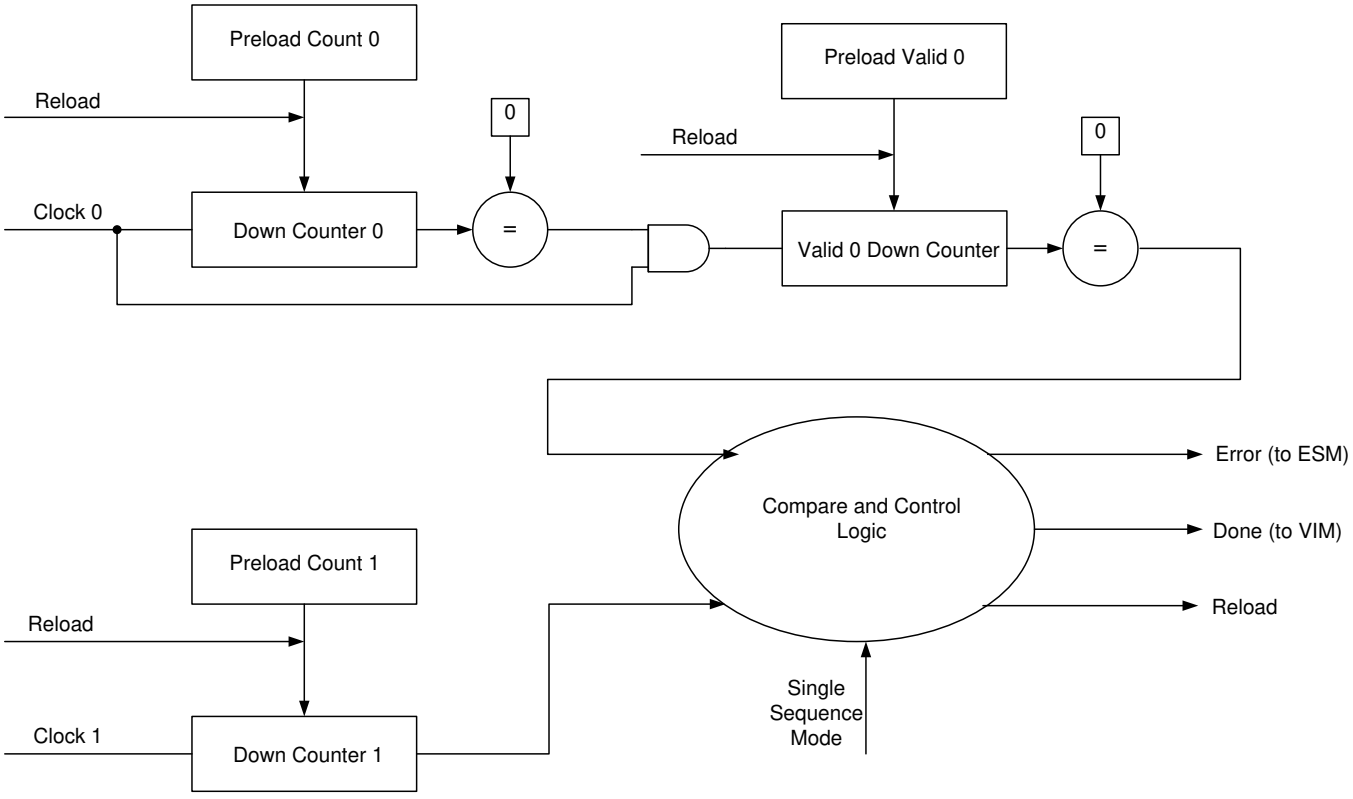
The main features of each of the DCC modules are:

- Allows application to ensure that a fixed ratio is maintained between frequencies of two clock signals
- Supports the definition of a programmable tolerance window in terms of number of reference clock cycles
- Supports continuous monitoring without requiring application intervention
- Also supports a single-sequence mode for spot measurements
- Allows selection of clock source for each of the counters resulting in several specific use cases

28.1.1.2 Block Diagram

Figure 28-1 illustrates the main concept of the DCC module.

Figure 28-1. DCC Operation



28.1.2 Module Operation

As shown in Figure 28-1, the DCC contains two counters – counter0 and counter1, which are driven by two signals – clock0 and clock1. The application programs the seed values for both these counters. The application also configures the tolerance window time by configuring the valid counter for clock0.

Counter0 and counter1 both start counting simultaneously once the DCC is enabled. When counter0 counts down to zero, this automatically triggers the count down of the tolerance window counter (valid0).

The DCC module can be used in two different operating modes:

28.1.2.1 Continuous Monitoring Mode

In this mode, the DCC is used by the application to ensure that two clock signals maintain the correct frequency ratio. Suppose the application wants to ensure that the PLL output signal (clock source # 1) always maintains a fixed frequency relationship with the main oscillator (clock source # 0).

- In this case, the application can use the main oscillator as the clock0 signal (for counter0 and valid0) and the PLL output as the clock1 (for counter1).
- The seed values of counter0, valid0 and counter1 are selected such that if the actual frequencies of clock0 and clock1 are equal to their expected frequencies, then the counter1 will reach zero either at the same time as counter0 or during the count down of the valid0 counter.
- If the counter1 reaches zero during the count down of the valid0 counter, then all the counters (counter0, valid0, counter1) are reloaded with their initial seed values once valid0 has also counted down to zero.
- This sequence of counting down and checking then continues as long as there is no error, or until the DCC module is disabled.
- The counters also all get reloaded if the application resets and restarts the DCC module.

Error Conditions:

An error condition is generated by any one of the following:

1. Counter1 counts down to 0 before Counter0 reaches 0. This means that clock1 is faster than expected, or clock0 is slower than expected. It includes the case when clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that clock1 is slower than expected. It includes the case when clock1 is stuck at 1 or 0.

   Any error freezes the counters from counting. An application may then read out the counter values to help determine what caused the error.

28.1.2.1.1 Error Conditions

While operating in continuous mode, the counters get reloaded with the seed values and continue counting down under the following conditions:

- The module is reset or restarted by the application, OR
- Counter0, Valid 0 and Counter1 all reach 0 without any error

Figure 28-2. Counter Relationship

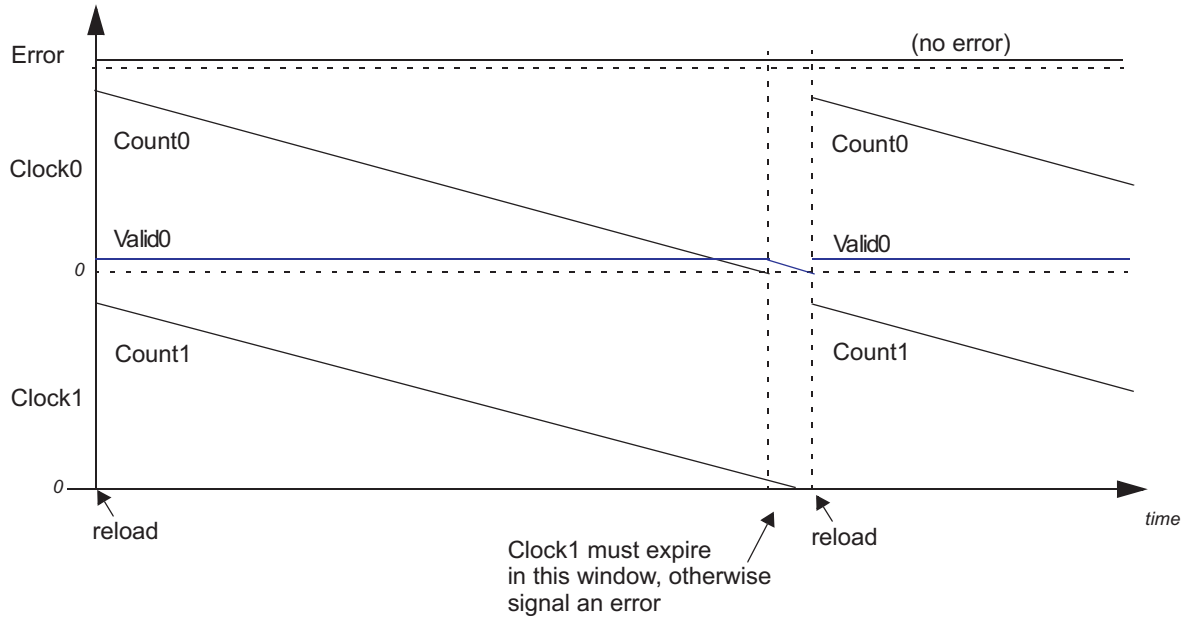


Figure 28-3. Clock1 Slower Than Clock0 - Results in an Error and Stops Counting

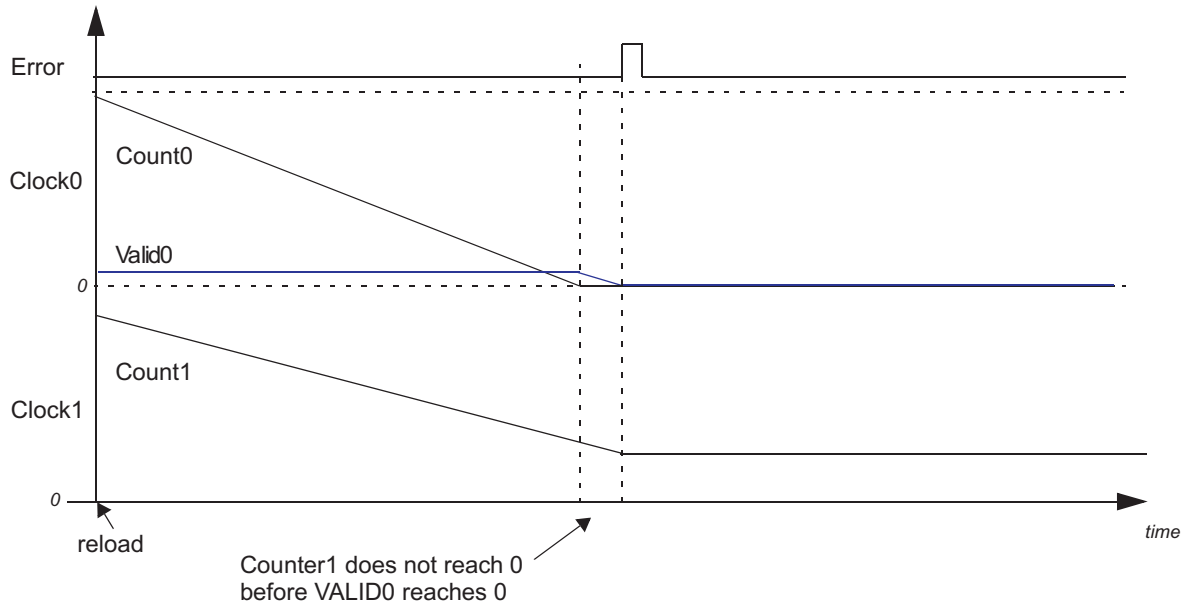


Figure 28-4. Clock1 Faster Than Clock0 - Results in an Error and Stops Counting

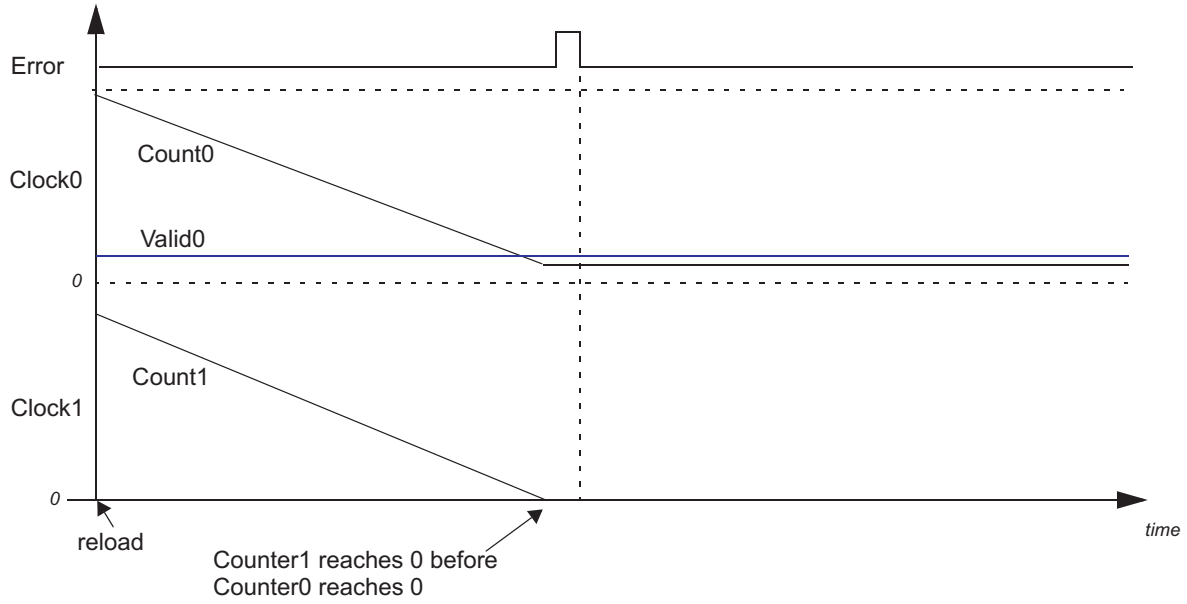


Figure 28-5. Clock1 Not Present - Results in an Error and Stops Counting

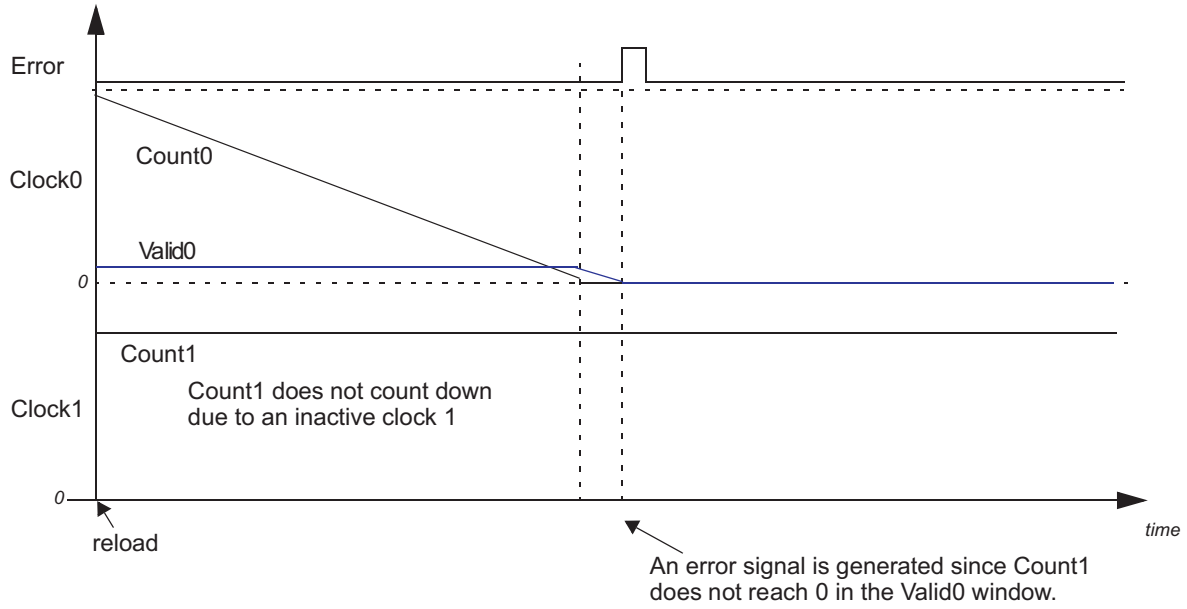
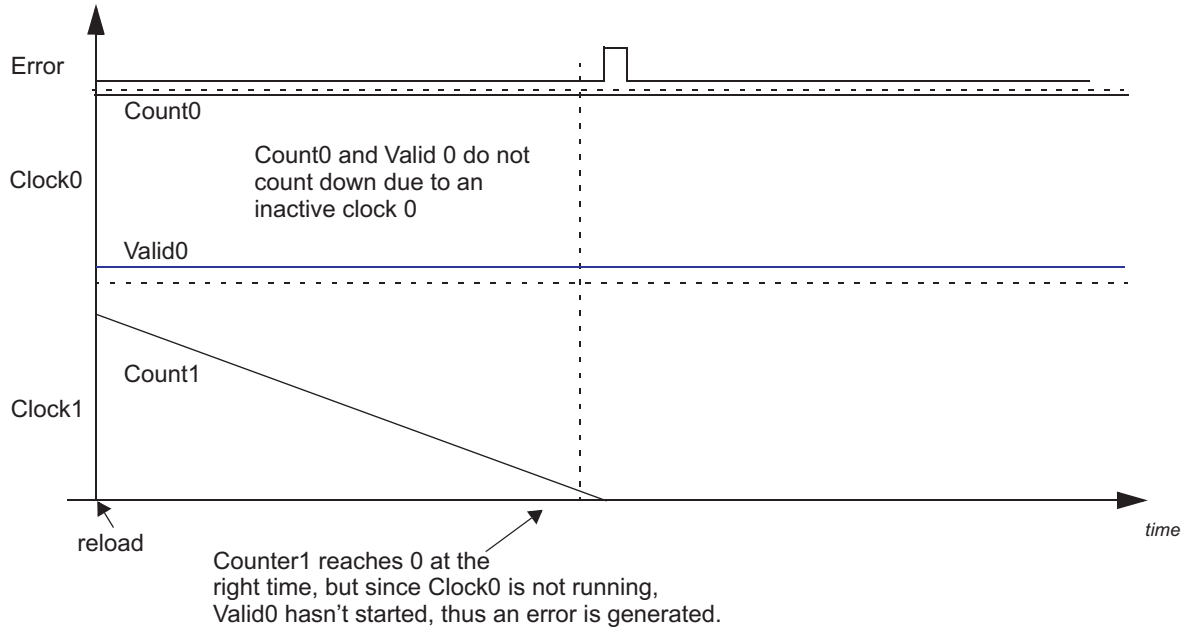


Figure 28-6. Clock0 Not Present - Results in an Error and Stops Counting



28.1.2.2 Single-Shot Measurement Mode

The DCC module can be programmed to count down one time by enabling the single-shot mode. In this mode, the DCC stops operating when the down counter0 and the valid counter0 reach 0. Alternatively, the DCC can be programmed to stop counting when the down counter1 reaches 0.

At the end of one sequence of counting down in this single-shot mode, the DCC gets disabled automatically, which prevents further counting. This mode is typically used for spot measurements of the frequency of a signal. This frequency could be an unknown for the application before the measurement.

Example Usage of Single-Shot Measurement Mode: Trimming the High-Frequency Low-Power Oscillator

A practical example of the usage of the spot measurement mode is in trimming the HF LPO (clock source # 5) using the main oscillator as a reference. This measurement sequence would proceed as follows:

- The application sets up the seed values for counter0 and valid0 for the duration of the measurement. Suppose the main oscillator frequency is 10MHz and the intended duration of the measurement is 500µs. The application needs to configure a seed value of 5000.
- These 5000 counts need to be divided between the counter0 and the valid0 counters. The minimum value for the valid0 seed is 4, so the application can configure counter0 seed value as 4996 and the valid0 seed value as 4.
- Suppose the HF LPO frequency is truly unknown. In this case the application can choose the maximum allowed seed value for counter1. This increases the probability of counter0 and valid0 counting down while the counter1 has still not fully counted down to zero. The maximum allowed seed value for counter1 is 1048575.
- Once the DCC is enabled, the counters counter0 and counter1 both start counting down from their seed values.
- When counter0 reaches zero, it automatically triggers the valid0 counter.
- When valid0 reaches zero, if counter1 is not zero as well, an ERROR status flag is set and a "DCC error" is sent to the ESM. Counter1 is also frozen so that it stops counting down any further. The application can enable an interrupt to be generated from the ESM whenever this DCC error is indicated. Refer the device datasheet to identify the ESM group and channel where the DCC error is connected.

- The DCC error interrupt service routine can then check the value of counter1 when the error was generated. Suppose that the counter1 now reads 1044575. This means that counter1 has counted 1048575 - 1044575, or 4000 cycles within the 500µs measurement period. This means that the average frequency of the HF LPO over this 500µs period was 4000 cycles / 500µs, or 8MHz.
- The application then needs to clear the ERROR status flag and restart the DCC module so that it is ready for the next spot measurement.

If there is no error generated at the end of the sequence, then the DONE status flag is set and a DONE interrupt is generated. The application must clear the DONE flag before restarting the DCC.

The conditions that cause a DCC error are identical between the continuous monitoring mode and the single-shot measurement mode.

Error Conditions:

An error condition is generated by any one of the following:

1. Counter1 counts down to 0 before Counter0 reaches 0. This means that clock1 is faster than expected, or clock0 is slower than expected. It includes the case when clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that clock1 is slower than expected. It includes the case when clock1 is stuck at 1 or 0.

   Any error freezes the counters from counting. An application may then read out the counter values to help determine what caused the error.

Freezing Counters when Counter1 Reaches Zero:

The DCC module also allows the counters to be frozen when the counter1 reaches zero. This allows one of the clock sources for counter1 to be used as a reference for measuring one of the clock sources for counter0. The error conditions are the same as those where (counter0=0 and valid0=0) define the condition when the DCC counters are frozen. That is, an error is indicated if coutner0 and valid0 become zero while counter1 is still non-zero. In this case, however, the application would typically set up the seed values such that the counter1 will become zero before counter0. Essentially the measurement period is defined by the seed value of the counter1. Note that this is also an error condition, and the interrupt service routine can use the measurement period and the actual cycles counted by counter1 to determine the frequency of the clock0 signal.

28.1.3 Clock Source Selection for Counter0 and Counter1

Refer the device datasheet to identify the available options for selecting the clock sources for both counters of the DCC module. Some microcontrollers may include multiple instances of the DCC module. This will also be identified in the device datasheet.

The selection of the clock sources for counter0 and coutner1 is done by a combination of the KEY, CNT0 CLKSRC and CNT1 CLKSRC control fields of the CNT0CLKSRC and CNT1CLKSRC registers.

28.1.4 DCC Registers

Table 28-1 lists the DCC memory-mapped registers. All register offset addresses not listed in Table 28-1 should be considered as reserved locations and the register contents should not be modified.

Table 28-1. DCC Registers

Offset	Acronym	Register Name	Section
0h	DCCGCTRL	DCCGCTRL	Section 28.1.4.1
4h	DCCREV	DCCREV	Section 28.1.4.2
8h	DCCCNTSEED0	DCCCNTSEED0	Section 28.1.4.3
Ch	DCCVALIDSEED0	DCCVALIDSEED0	Section 28.1.4.4
10h	DCCCNTSEED1	DCCCNTSEED1	Section 28.1.4.5
14h	DCCSTAT	DCCSTAT	Section 28.1.4.6
18h	DCCCNT0	DCCCNT0	Section 28.1.4.7
1Ch	DCCVALID0	DCCVALID0	Section 28.1.4.8
20h	DCCCNT1	DCCCNT1	Section 28.1.4.9
24h	DCCCLKSSRC1	DCCCLKSSRC1	Section 28.1.4.10
28h	DCCCLKSSRC0	DCCCLKSSRC0	Section 28.1.4.11

Complex bit access types are encoded to fit into small table cells. Table 28-2 shows the codes that are used for access types in this section.

Table 28-2. DCC Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

28.1.4.1 DCCGCTRL Register (Offset = 0h) [reset = 5555h]

DCCGCTRL is shown in Figure 28-7 and described in Table 28-3.

Return to Summary Table.

Starts / stops the counters clears the error signal

Figure 28-7. DCCGCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DONENA				SINGLESHOT				ERRENA				DCCENA			
R/W-5h				R/W-5h				R/W-5h				R/W-5h			

Table 28-3. DCCGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	Reserved
15-12	DONENA	R/W	5h	The DONEENA bit enables/disables the done signal. 0101 = disabled & 1010 = enabled
11-8	SINGLESHOT	R/W	5h	Single/Continuous checking mode. 0101 = Continuous & 1010 = Single
7-4	ERRENA	R/W	5h	The ERRENA bit enables/disables the error signal. 0101 = disabled & 1010 = enabled
3-0	DCCENA	R/W	5h	The DCCENA bit starts and stops the operation of the dcc 0101 = disabled & 1010 = enabled

28.1.4.2 DCCREV Register (Offset = 4h) [reset = 40000204h]

DCCREV is shown in Figure 28-8 and described in Table 28-4.

Return to Summary Table.

Module version

Figure 28-8. DCCREV Register

31	30	29	28	27	26	25	24
NU2	SCHEME			NU1		FUNC	
R-0h	R-4h			R-0h		R-0h	
23	22	21	20	19	18	17	16
FUNC							
R-0h							
15	14	13	12	11	10	9	8
FUNC		RTL					MAJOR
R-0h		R-1h					R-0h
7	6	5	4	3	2	1	0
MAJOR		CUSTOM	MINOR				
R-0h		R-0h	R-4h				

Table 28-4. DCCREV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU2	R	0h	Reserved
30-28	SCHEME	R	4h	SCHEME. - (RO)
27-26	NU1	R	0h	Reserved
25-14	FUNC	R	0h	Functional release number - (RO)
13-9	RTL	R	1h	Design Release Number - (RO)
8-6	MAJOR	R	0h	Major Revision Number - (RO)
5	CUSTOM	R	0h	Indicates a special version of the module. May not be supported by standard software - (RO)
4-0	MINOR	R	4h	Minor revision number. - (RO)

28.1.4.3 DCCCNTSEED0 Register (Offset = 8h) [reset = 0h]

DCCCNTSEED0 is shown in Figure 28-9 and described in Table 28-5.

Return to Summary Table.

Seed value for the counter attached to clock source 0

Figure 28-9. DCCCNTSEED0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3												COUNTSEED0																			
R-0h												R/W-0h																			

Table 28-5. DCCCNTSEED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU3	R	0h	Reserved
19-0	COUNTSEED0	R/W	0h	The seed value for Counter 0. The seed value that gets loaded into counter 0 (clock source 0)

28.1.4.4 DCCVALIDSEED0 Register (Offset = Ch) [reset = 0h]

DCCVALIDSEED0 is shown in Figure 28-10 and described in Table 28-6.

Return to Summary Table.

Seed value for the timeout counter attached to clock source 0

Figure 28-10. DCCVALIDSEED0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU4																VALIDSEED0															
R-0h																R/W-0h															

Table 28-6. DCCVALIDSEED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU4	R	0h	Reserved
15-0	VALIDSEED0	R/W	0h	The seed value for Valid Duration Counter 0.The seed value that gets loaded into the valid duration counter for clock source 0

28.1.4.5 DCCCNTSEED1 Register (Offset = 10h) [reset = 0h]

DCCCNTSEED1 is shown in Figure 28-11 and described in Table 28-7.

Return to Summary Table.

Seed value for the counter attached to clock source 1

Figure 28-11. DCCCNTSEED1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU5												COUNTSEED1																			
R-0h												R/W-0h																			

Table 28-7. DCCCNTSEED1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU5	R	0h	Reserved
19-0	COUNTSEED1	R/W	0h	The seed value for Counter 1. The seed value that gets loaded into counter 1 (clock source 1

28.1.4.6 DCCSTAT Register (Offset = 14h) [reset = 0h]

DCCSTAT is shown in Figure 28-12 and described in Table 28-8.

Return to Summary Table.

Contains the error & done flag bit

Figure 28-12. DCCSTAT Register

31	30	29	28	27	26	25	24
NU6							
R-0h							
23	22	21	20	19	18	17	16
NU6							
R-0h							
15	14	13	12	11	10	9	8
NU6							
R-0h							
7	6	5	4	3	2	1	0
NU6						DONE	ERR
R-0h						R/W-0h	R/W-0h

Table 28-8. DCCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU6	R	0h	Reserved
1	DONE	R/W	0h	Indicates whether or not an done has occured. Writing a 1 to this bit clears the flag.
0	ERR	R/W	0h	Indicates whether or not an error has occured. Writing a 1 to this bit clears the flag.

28.1.4.7 DCCCNT0 Register (Offset = 18h) [reset = 0h]

DCCCNT0 is shown in Figure 28-13 and described in Table 28-9.

Return to Summary Table.

Value of the counter attached to clock source 0

Figure 28-13. DCCCNT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU7												COUNT0																			
R-0h												R-0h																			

Table 28-9. DCCCNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU7	R	0h	Reserved
19-0	COUNT0	R	0h	This field contains the current value of counter 0. - (RO)

28.1.4.8 DCCVALID0 Register (Offset = 1Ch) [reset = 0h]

DCCVALID0 is shown in Figure 28-14 and described in Table 28-10.

Return to Summary Table.

Value of the valid counter attached to clock source 0

Figure 28-14. DCCVALID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU8																VALID0															
R-0h																R-0h															

Table 28-10. DCCVALID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU8	R	0h	Reserved
15-0	VALID0	R	0h	This field contains the current value of valid counter 0. - (RO)

28.1.4.9 DCCCNT1 Register (Offset = 20h) [reset = 0h]

DCCCNT1 is shown in Figure 28-15 and described in Table 28-11.

Return to Summary Table.

Value of the counter attached to clock source 1

Figure 28-15. DCCCNT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU9												COUNT1																			
R-0h												R-0h																			

Table 28-11. DCCCNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU9	R	0h	Reserved
19-0	COUNT1	R	0h	This field contains the current value of counter 1. - (RO)

28.1.4.10 DCCCLKSSRC1 Register (Offset = 24h) [reset = 5000h]

DCCCLKSSRC1 is shown in Figure 28-16 and described in Table 28-12.

Return to Summary Table.

Clock source1 selection control

Figure 28-16. DCCCLKSSRC1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU11															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY_B4				NU10								CLK_SRC1			
R/W-5h				R-0h								R/W-0h			

Table 28-12. DCCCLKSSRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU11	R	0h	Reserved
15-12	KEY_B4	R/W	5h	Key Programing (1010 is the KEY Value)
11-4	NU10	R	0h	Reserved
3-0	CLK_SRC1	R/W	0h	Clock source selection for Source 0 DCC-A Clock source-0 selection Program value and its respective clock selected 0x0 - REF_CLK 0x1 - CPU_CLK 0x2 - RC_CLK 0x3 - RC_CLK 0x4 - RC_CLK 0x5 - RC_CLK 0x6 - RC_CLK 0x7 - RC_CLK DCC-B Clock source-0 selection Program value and its respective clock selected 0x0 - VCLK 0x1 - DSS_CLK 0x2 - BSS_CLK 0x3 - QSPI_CLK 0x4 - FDCAN_CLK 0x5 - RED_CLK 0x6 - CPU_CLK 0x7 - RC_CLK

28.1.4.11 DCCCLKSSRC0 Register (Offset = 28h) [reset = 5h]

DCCCLKSSRC0 is shown in Figure 28-17 and described in Table 28-13.

Return to Summary Table.

Clock source0 selection control

Figure 28-17. DCCCLKSSRC0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU12															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU12												CLK_SRC0			
R-0h												R/W-5h			

Table 28-13. DCCCLKSSRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU12	R	0h	Reserved
3-0	CLK_SRC0	R/W	5h	Clock source selection for Source 0 DCC-A Clock source-0 selection Program value and its respective clock selected 0 - REF_CLK A - PLL_600 5 - PLL_240 DCC-B Clock source-0 selection Program value and its respective clock selected 0 - PLL_600 A - VCLK 5 - CPU_CLK

28.2 Error Signaling Module (ESM)

This section provides the details of the error signaling module (ESM) that aggregates device errors and provides internal and external error response based on error severity.

28.2.1 Overview

The Error Signaling Module (ESM) collects and reports the various error conditions on the microcontroller. The error condition is categorized based on a severity level. Error response is then generated based on the category of the error. Possible error responses include a low priority interrupt, high priority interrupt, and an external pin action.

28.2.1.1 Feature List

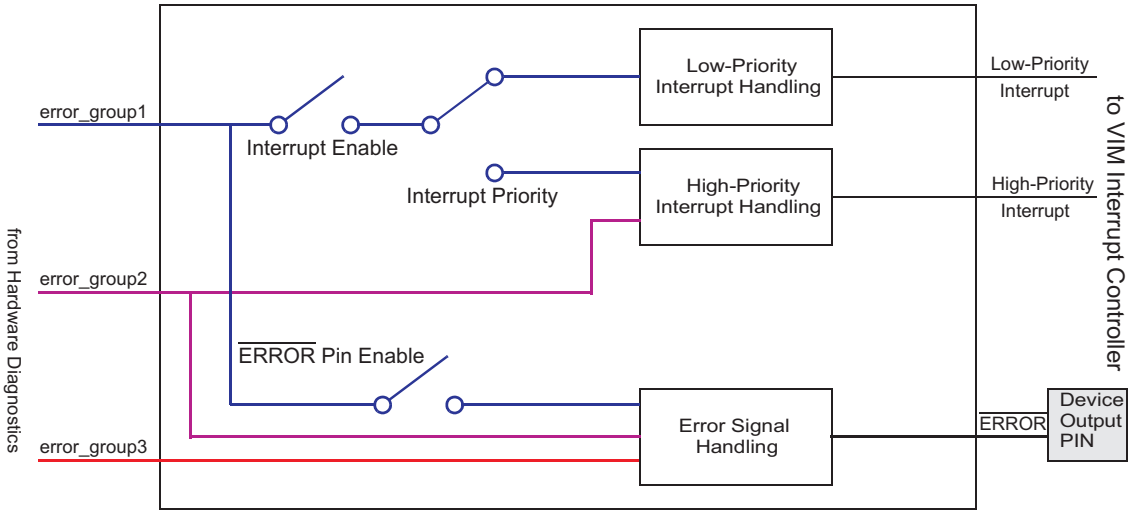
- Up to 128 error channels are supported, divided into 3 different groups:
  - 64 Group1 (low severity) channels with configurable interrupt generation and configurable $\overline{\text{ERROR}}$ pin behavior
  - 32 Group2 (high severity) channels with predefined interrupt generation and predefined $\overline{\text{ERROR}}$ pin behavior
  - 32 Group3 (high severity) channels with no interrupt generation and predefined $\overline{\text{ERROR}}$ pin behavior. These channels have no interrupt response as they are reserved for CPU based diagnostics that generate aborts directly to the CPU.
- Dedicated device $\overline{\text{ERROR}}$ pin to signal an external observer
- Configurable timebase for $\overline{\text{ERROR}}$ pin output
- Error forcing capability for latent fault testing

28.2.1.2 Block Diagram

As shown in Figure 28-18, the ESM channels are divided into three groups. Group1 channels are considered to be low severity. Group1 errors have a configurable interrupt response and configurable $\overline{\text{ERROR}}$ pin behavior. Note that the ESM Status Register 1 (ESMSR1) for error group 1 gets updated, regardless if the interrupt enable is active or not. Group2 channels are $\overline{\text{ERROR}}$ high severity. Group2 errors always generate a high priority interrupt and an output on the $\overline{\text{ERROR}}$ pin. Group3 errors are reserved for high severity errors generated by diagnostics which have already generated a CPU abort response. Because an abort response is generated, there is no need to generate an interrupt response. Group3 errors always generate an $\overline{\text{ERROR}}$ pin output.

The ESM interrupt and $\overline{\text{ERROR}}$ pin behavior are also summarized in Table 28-14.

Figure 28-18. Block Diagram



Note that the ESM Status Register 1 (ESMSR1) for error_group1 gets updated, regardless if the interrupt enable is active or not.

Table 28-14. ESM Interrupt and $\overline{\text{ERROR}}$ Pin Behavior

Error Group	Interrupt Generated	Interrupt Priority	$\overline{\text{ERROR}}$ Pin Response Generated
1	configurable interrupt	configurable priority	configurable output generation
2	interrupt generated	high priority	output generated
3	no interrupt	NA	output generated

Figure 28-19 and Figure 28-20 show the interrupt response handling and $\overline{\text{ERROR}}$ pin response handling with register configuration. The total active time of the $\overline{\text{ERROR}}$ pin is controlled by the Low-Time Counter Preload register (LTCP) and the key register (ESMEPSR) as shown in Figure 28-20. See Section 28.2.2.2 for details.

Figure 28-19. Interrupt Response Handling

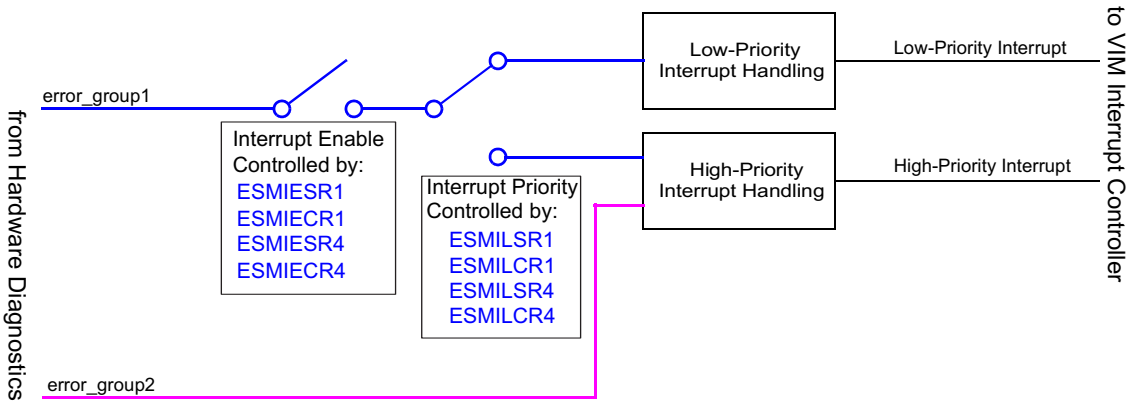
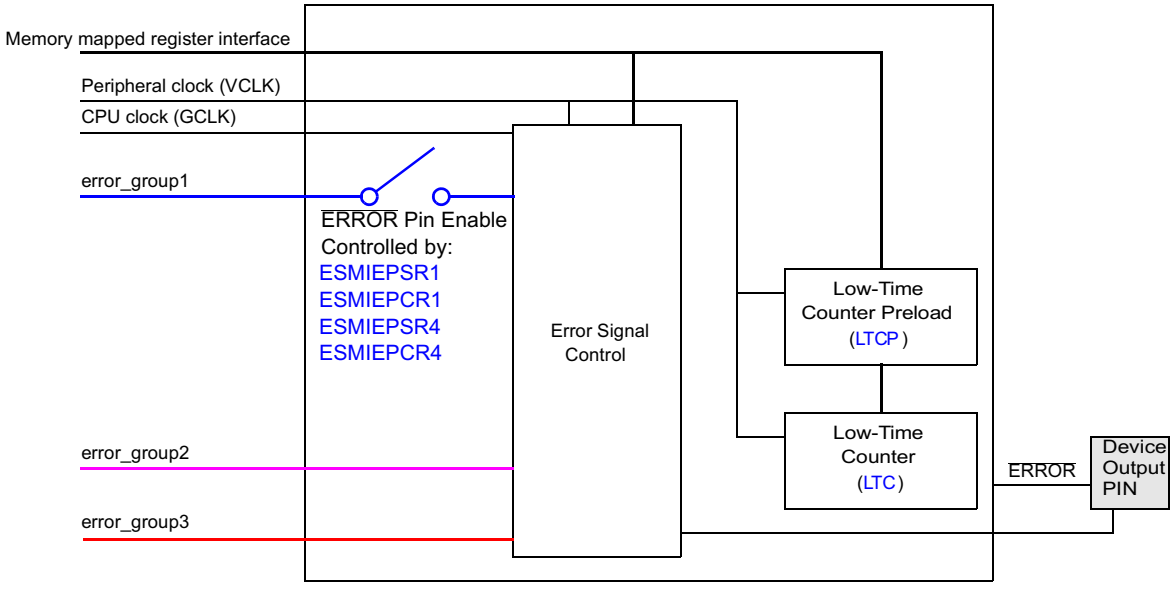


Figure 28-20. $\overline{\text{ERROR}}$ Pin Response Handling



28.2.2 Module Operation

This device has 128 error channels, divided into 3 different error groups. Please refer to the device datasheet for ESM channel assignment details.

The ESM module has error flags for each error channel. The error status registers ESMSR1, ESMSR4, ESMSR2, ESMSR3 provide status information on a pending error of Group1 (Channel 0-31), Group1 (Channel 32-63), Group1 (Channel 64-95), Group2, and Group3, respectively. The ESMEPSR register provides the current $\overline{\text{ERROR}}$ status. The module also provides a status shadow register, ESMSSR2, which maintains the error flags of Group2 until power-on reset ($\overline{\text{PORRST}}$) is asserted. See Section 28.2.2.1 for details of their behavior during power on reset and warm reset.

Once an error occurs, the ESM module will set the corresponding error flags. In addition, it can trigger an interrupt, $\overline{\text{ERROR}}$ pin outputs low depending on the ESM settings. Once the $\overline{\text{ERROR}}$ pin outputs low, a power on reset or a write of 0x5 to ESMEKR is required to release the ESM error pin back to normal state. See Section 28.2.2.2 for details. The application can read the error status registers (ESMSR1, ESMSR4, ESMSR7, ESMSR2, and ESMSR3) to debug the error. If an $\overline{\text{RST}}$ is triggered or the error interrupt has been served, the error flag of Group2 should be read from ESMSSR2 because the error flag in ESMSR2 will be cleared by $\overline{\text{RST}}$.

You can also test the functionality of the $\overline{\text{ERROR}}$ pin by forcing an error. See Section 28.2.2.3 for details.

28.2.2.1 Reset Behavior

Power on reset:

- $\overline{\text{ERROR}}$ pin behavior

  When nPORRST is active, the $\overline{\text{ERROR}}$ pin is in a high impedance state (output drivers disabled).
- Register behavior

  After $\overline{\text{PORRST}}$, all registers in ESM module will be re-initialized to the default value. All the error status registers are cleared to zero.

Warm reset ($\overline{\text{RST}}$):

- $\overline{\text{ERROR}}$ pin behavior

  During $\overline{\text{RST}}$, the $\overline{\text{ERROR}}$ pin is in “output active” state with pull-down disabled. The $\overline{\text{ERROR}}$ pin remains unchanged after $\overline{\text{RST}}$.
- Register behavior

  After $\overline{\text{RST}}$, ESMSR1, ESMSR4, ESMSR7, ESMSSR2, ESMSR3 and ESMEPSR register values remains un-changed. Since $\overline{\text{RST}}$ does not clear the critical failure registers, the user can read those registers to debug the failures after $\overline{\text{RST}}$ pin goes back to high.

  After $\overline{\text{RST}}$, if one of the flags in ESMSR1, ESMSR4 and ESMSR7 is set, the interrupt service routine will be called once the corresponding interrupt is enabled.

NOTE: ESMSR2 is cleared after $\overline{\text{RST}}$. The flag in ESMSR2 gets cleared when reading the appropriate vector in the ESMIOFFHR offset register. Reading ESMIOFFHR will not clear the ESMSR1, ESMSR4, ESMSR7 and the shadow register ESMSSR2. Reading ESMIOFFLR will also not clear the ESMSR1, ESMSR4 and ESMSR7.

28.2.2.2 $\overline{\text{ERROR}}$ Pin Timing

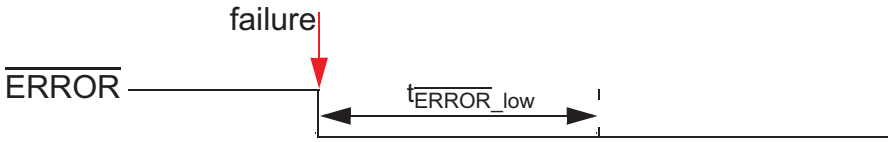
The $\overline{\text{ERROR}}$ pin is an active low function. The state of the pin is also readable from $\overline{\text{ERROR}}$ Pin Status Register (ESMEPSR). A warm reset ($\overline{\text{RST}}$) does not affect the state of the pin. The pin is in a high-impedance state during power-on reset. Once the ESM module drives the $\overline{\text{ERROR}}$ pin low, it remains in this state for the time specified by the Low-Time Counter Preload register (LTCPR). Based on the time period of the peripheral clock (VCLK), the total active time of the $\overline{\text{ERROR}}$ pin can be calculated as:

$$t_{\overline{ERROR}\_low} = t_{VCLK} \times (LTCP + 1) \quad (21)$$

Once this period expires, the $\overline{\text{ERROR}}$ pin is set to high in case the reset of the $\overline{\text{ERROR}}$ pin was requested. This request is done by writing an appropriate key (0x5) to the key register (ESMEKR) during the $\overline{\text{ERROR}}$ pin low time. Here are a few examples:

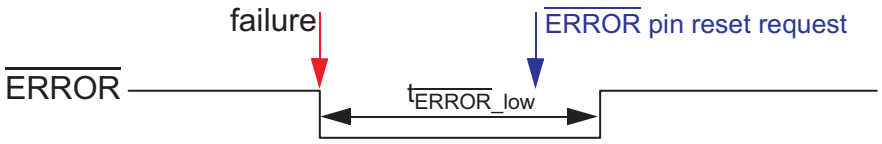
Example 1: ESM detects a failure and drives the $\overline{\text{ERROR}}$ pin low. No $\overline{\text{ERROR}}$ pin reset is requested. The $\overline{\text{ERROR}}$ pin continues outputting low until power on reset occurs.

Figure 28-21. $\overline{\text{ERROR}}$ Pin Timing - Example 1



Example 2: ESM detects a failure and drives the $\overline{\text{ERROR}}$ pin low. An $\overline{\text{ERROR}}$ pin reset request is received before $t_{\overline{ERROR}\_low}$ expires. In this case, the $\overline{\text{ERROR}}$ pin is set to high immediately after $t_{\overline{ERROR}\_low}$ expires.

Figure 28-22. $\overline{\text{ERROR}}$ Pin Timing - Example 2

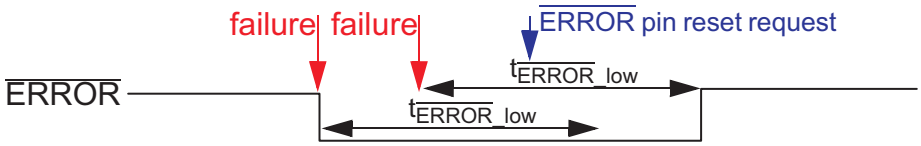


Example 3: ESM detects a failure and drives the $\overline{\text{ERROR}}$ pin low. An $\overline{\text{ERROR}}$ pin reset request is received after $t_{\overline{ERROR}\_low}$ expires. In this case, the $\overline{\text{ERROR}}$ pin is set to high immediately after $\overline{\text{ERROR}}$ pin reset request is received.

Figure 28-23. $\overline{\text{ERROR}}$ Pin Timing - Example 3

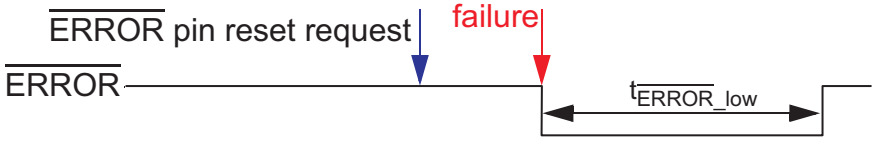
Example 4: ESM detects a failure and drives the $\overline{\text{ERROR}}$ pin low. Another failure occurs within the time the pin stays low. In this case, the low time counter will be reset when the other failure occurs. In other words, $t_{\overline{\text{ERROR}}\_low}$ should be counted from whenever the most recent failure occurs.

Figure 28-24. $\overline{\text{ERROR}}$ Pin Timing - Example 4



Example 5: The reset of the $\overline{\text{ERROR}}$ pin was requested by the software even before the failure occurs. In this case, the $\overline{\text{ERROR}}$ pin is set to high immediately after $t_{\overline{\text{ERROR}}\_low}$ expires. This case is not recommended and should be avoided by the application.

Figure 28-25. $\overline{\text{ERROR}}$ Pin Timing - Example 5



28.2.2.3 Forcing an Error Condition

The error response generation mechanism is testable by software by forcing an error condition. This allows testing the $\overline{\text{ERROR}}$ pin functionality. By writing a dedicated key to the error forcing key register (ESMEKR), the $\overline{\text{ERROR}}$ pin is set to low for the specified time. The following steps describe how to force an error condition:

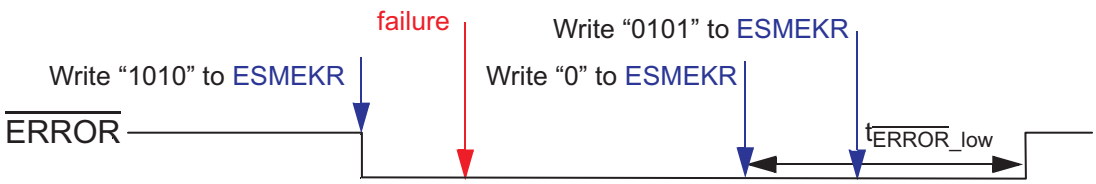
1. Check $\overline{\text{ERROR}}$ Pin Status Register (ESMEPSR). This register must be 1 to switch into the error forcing mode.

   The ESM module cannot be switched into the error forcing mode if a failure has already been detected in functional mode. The application command to switch to error forcing mode is ignored.
2. Write “1010b” to the error forcing key register (ESMEKR). After that, the $\overline{\text{ERROR}}$ pin should output low (error force mode).

   Once the application puts the ESM module in the error forcing mode, the $\overline{\text{ERROR}}$ pin cannot indicate the normal error functionality. If a failure occurs during this time, it gets still latched and the LTC is reset and stopped. The error output pin is already driven low on account of the error forcing mode. When the ESM is forced back to normal functional mode, the LTC becomes active and forces the $\overline{\text{ERROR}}$ pin low until the expiration of the LTC (see Figure 28-26).
3. Write “0000” to the error forcing key register (ESMEKR) back to the active normal mode.

   If there are no errors detected while the ESM module is in the error forcing mode, the $\overline{\text{ERROR}}$ pin goes high immediately after exiting the error forcing mode.

Figure 28-26. ERROR Pin Timing - Example 6

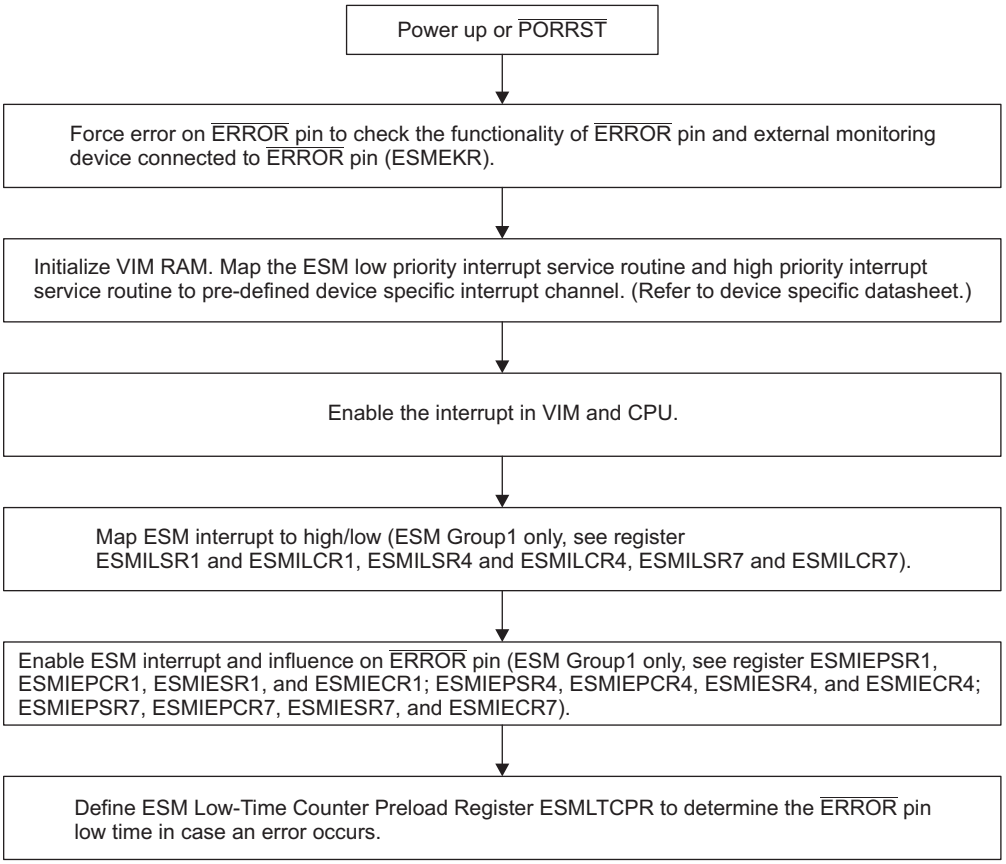


28.2.3 Recommended Programming Procedure

During the initialization stage, the application code should follow the recommendations in Figure 28-27 to initialize the ESM.

Once an error occurs, it can trigger an interrupt, $\overline{\text{ERROR}}$ pin outputs low depending on the ESM settings. Once the $\overline{\text{ERROR}}$ pin outputs low, a power on reset or a write of 0x5 to ESMEKR is required to release the ESM back to normal state. The application can read the error status registers (ESMSR1, ESMSR4, ESMSR7, ESMSR2, and ESMSR3) to debug the error. If an $\overline{\text{RST}}$ is triggered or the error interrupt has been served, the error flag of Group2 should be read from ESMSSR2 because the error flag in ESMSR2 will be cleared by $\overline{\text{RST}}$.

Figure 28-27. ESM Initialization



28.2.4 MSS_ESM Registers

Table 28-15 lists the memory-mapped registers for the MSS_ESM. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

Table 28-15. MSS_ESM Registers

Offset	Acronym	Register Name	Section
0h	ESMIEPSR1	ESM Enable ERROR Pin Action/Response Register 1	Section 28.2.4.1
4h	ESMIEPCR1	ESM Disable ERROR Pin Action/Response Register 1	Section 28.2.4.2
8h	ESMIESR1	ESM Interrupt Enable Set/Status Register 1	Section 28.2.4.3
Ch	ESMIECR1	ESM Interrupt Enable Clear/Status Register 1	Section 28.2.4.4
10h	ESMILSR1	Interrupt Level Set/Status Register 1	Section 28.2.4.5
14h	ESMILCR1	Interrupt Level Clear/Status Register 1	Section 28.2.4.6
18h	ESMSR1	ESM Status Register 1	Section 28.2.4.7
1Ch	ESMSR2	ESM Status Register 2	Section 28.2.4.8
20h	ESMSR3	ESM Status Register 3	Section 28.2.4.9
24h	ESMEPSR	ESM ERROR Pin Status Register	Section 28.2.4.10
28h	ESMIOFFHR	ESM Interrupt Offset High Register	Section 28.2.4.11
2Ch	ESMIOFFLR	ESM Interrupt Offset Low Register	Section 28.2.4.12
30h	ESMLTCR	ESM Low-Time Counter Register	Section 28.2.4.13
34h	ESMLTCPR	ESM Low-Time Counter Preload Register	Section 28.2.4.14
38h	ESMEKR	ESM Error Key Register	Section 28.2.4.15
3Ch	ESMSSR2	ESM Status Shadow Register 2	Section 28.2.4.16
40h	ESMIEPSR4	ESM Enable ERROR Pin Action/Response Register 4	Section 28.2.4.17
44h	ESMIEPCR4	ESM Disable ERROR Pin Action/Response Register 4	Section 28.2.4.18
48h	ESMIESR4	ESM Interrupt Enable Set/Status Register 4	Section 28.2.4.19
4Ch	ESMIECR4	ESM Interrupt Enable Clear/Status Register 4	Section 28.2.4.20
50h	ESMILSR4	Interrupt Level Set/Status Register 4	Section 28.2.4.21
54h	ESMILCR4	Interrupt Level Clear/Status Register 4	Section 28.2.4.22
58h	ESMSR4	ESM Status Register 4	Section 28.2.4.23
80h	ESMIEPSR7	ESM Enable ERROR Pin Action/Response Register 7	Section 28.2.4.24
84h	ESMIEPCR7	ESM Disable ERROR Pin Action/Response Register 7	Section 28.2.4.25
88h	ESMIESR7	ESM Interrupt Enable Set/Status Register 7	Section 28.2.4.26
8Ch	ESMIECR7	ESM Interrupt Enable Clear/Status Register 7	Section 28.2.4.27
90h	ESMILSR7	Interrupt Level Set/Status Register 7	Section 28.2.4.28
94h	ESMILCR7	Interrupt Level Clear/Status Register 7	Section 28.2.4.29
98h	ESMSR7	ESM Status Register 7	Section 28.2.4.30

Complex bit access types are encoded to fit into small table cells. Table 28-16 shows the codes that are used for access types in this section.

Table 28-16. MSS_ESM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

28.2.4.1 ESMIEPSR1 Register (Offset = 0h) [reset = 0h]

ESMIEPSR1 is shown in Figure 28-28 and described in Table 28-17.

Return to Summary Table.

ESM Enable ERROR Pin Action/Response Register 1

Figure 28-28. ESMIEPSR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPSET																															
R/W-0h																															

Table 28-17. ESMIEPSR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IEPSET	R/W	0h	Enable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding clear bit in the ESMIEPCR1 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Enables failure influence on ERROR pin and sets the corresponding clear bit in the ESMIEPCR1 register.

28.2.4.2 ESMIEPCR1 Register (Offset = 4h) [reset = 0h]

ESMIEPCR1 is shown in Figure 28-29 and described in Table 28-18.

Return to Summary Table.

ESM Disable ERROR Pin Action/Response Register 1

Figure 28-29. ESMIEPCR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPCLR																															
R/W-0h																															

Table 28-18. ESMIEPCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IEPCLR	R/W	0h	Disable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding set bit in the ESMIEPSR1 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Disables failure influence on ERROR pin and clears the corresponding set bit in the ESMIEPSR1 register.

28.2.4.3 ESMIESR1 Register (Offset = 8h) [reset = 0h]

ESMIESR1 is shown in Figure 28-30 and described in Table 28-19.

Return to Summary Table.

ESM Interrupt Enable Set/Status Register 1

Figure 28-30. ESMIESR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSET																															
R/W-0h																															

Table 28-19. ESMIESR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTENSET	R/W	0h	Set interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding clear bit in the ESMIECR1 register unchanged. 1 Read: Interrupt is enabled. Write: Enables interrupt and sets the corresponding clear bit in the ESMIECR1 register.

28.2.4.4 ESMIECR1 Register (Offset = Ch) [reset = 0h]

ESMIECR1 is shown in Figure 28-31 and described in Table 28-20.

Return to Summary Table.

ESM Interrupt Enable Clear/Status Register 1

Figure 28-31. ESMIECR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENCLR																															
R/W-0h																															

Table 28-20. ESMIECR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTENCLR	R/W	0h	Clear Interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding set bit in the ESMIESR1 register unchanged. 1 Read: Interrupt is enabled. Write: Disables interrupt and clears the corresponding set bit in the ESMIESR1 register.

28.2.4.5 ESMILSR1 Register (Offset = 10h) [reset = 0h]

ESMILSR1 is shown in Figure 28-32 and described in Table 28-21.

Return to Summary Table.

Interrupt Level Set/Status Register 1

Figure 28-32. ESMILSR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLSET																															
R/W-0h																															

Table 28-21. ESMILSR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLVLSET	R/W	0h	Set Interrupt Priority Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding clear bit in the ESMILCR1 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to high level interrupt line and sets the corresponding clear bit in the ESMILCR1 register.

28.2.4.6 ESMILCR1 Register (Offset = 14h) [reset = 0h]

ESMILCR1 is shown in Figure 28-33 and described in Table 28-22.

Return to Summary Table.

Interrupt Level Clear/Status Register 1

Figure 28-33. ESMILCR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLCLR																															
R/W-0h																															

Table 28-22. ESMILCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLVLCLR	R/W	0h	Clear Interrupt Priority. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding set bit in the ESMILSR1 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to low level interrupt line and clears the corresponding set bit in the ESMILSR1 register.

28.2.4.7 ESMSR1 Register (Offset = 18h) [reset = 0h]

ESMSR1 is shown in Figure 28-34 and described in Table 28-23.

Return to Summary Table.

ESM Status Register 1

Figure 28-34. ESMSR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

Table 28-23. ESMSR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. Note: After nRST, if one of these flags are set and the corresponding interrupt are enabled, the interrupt service routine will be called.

28.2.4.8 ESMSR2 Register (Offset = 1Ch) [reset = 0h]

ESMSR2 is shown in Figure 28-35 and described in Table 28-24.

Return to Summary Table.

ESM Status Register 2

Figure 28-35. ESMSR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

Table 28-24. ESMSR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. ESMSSR2 is not impacted by this action. Note: In normal operation the flag gets cleared when reading the appropriate vector in the ESMIOFFHR offset register. Reading ESMIOFFHR will not clear the ESMSR1 and the shadow register ESMSSR2.

28.2.4.9 ESMSR3 Register (Offset = 20h) [reset = 0h]

ESMSR3 is shown in Figure 28-36 and described in Table 28-25.

Return to Summary Table.

ESM Status Register 3

Figure 28-36. ESMSR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

Table 28-25. ESMSR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred. Write: Leaves the bit unchanged. 1 Read: Error occurred. Write: Clears the bit.

28.2.4.10 ESMEPSR Register (Offset = 24h) [reset = 0h]

ESMEPSR is shown in Figure 28-37 and described in Table 28-26.

Return to Summary Table.

ESM ERROR Pin Status Register

Figure 28-37. ESMEPSR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							EPSF
R/W-0h							R/W-0h

Table 28-26. ESMEPSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
0	EPSF	R/W	0h	ERROR Pin Status Flag. Provides status information for the ERROR Pin. Read/Write in User and Privileged mode. 0 Read: ERROR Pin is low (active) if any error has occurred. Write: Writes have no effect. 1 Read: ERROR Pin is high if no error has occurred. Write: Writes have no effect. Note: This flag will be set to 1 after PORRST. The value will be unchanged after nRST. The ERROR pin status remains un-changed during after nRST.

28.2.4.11 ESMIOFFHR Register (Offset = 28h) [reset = 0h]

ESMIOFFHR is shown in Figure 28-38 and described in Table 28-27.

Return to Summary Table.

ESM Interrupt Offset High Register

Figure 28-38. ESMIOFFHR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									INTOFFH						
R/W-0h																									R/W-0h						

Table 28-27. ESMIOFFHR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
6-0	INTOFFH	R/W	0h	Offset High Level Interrupt. This vector gives the channel number of the highest pending interrupt request for the high level interrupt line. Interrupts of error Group2 have higher priority than interrupts of error Group1. Inside a group, channel 0 has highest priority and channel 31 has lowest priority. User and privileged mode (read): Returns number of pending interrupt with the highest priority for the high level interrupt line. 0 No pending interrupt. 1h Interrupt pending for channel 0, error Group1. ... 20h Interrupt pending for channel 31, error Group1. 21h Interrupt pending for channel 0, error Group2. ... 40h Interrupt pending for channel 31, error Group2. 41h Interrupt pending for channel 32, error Group1. ... 60h Interrupt pending for channel 63, error Group1. Note: Reading the interrupt vector will clear the corresponding flag in the ESMSR2 register; will not clear ESMSR1 and ESMSSR2 and the offset register gets updated. User and privileged mode (write): Writes have no effect.

28.2.4.12 ESMIOFFLR Register (Offset = 2Ch) [reset = 0h]

ESMIOFFLR is shown in Figure 28-39 and described in Table 28-28.

Return to Summary Table.

ESM Interrupt Offset Low Register

Figure 28-39. ESMIOFFLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									INTOFFL						
R/W-0h																									R/W-0h						

Table 28-28. ESMIOFFLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
6-0	INTOFFL	R/W	0h	Offset Low Level Interrupt. This vector gives the channel number of the highest pending interrupt request for the low level interrupt line. Inside a group, channel 0 has highest priority and channel 31 has lowest priority. User and privileged mode (read): Returns number of pending interrupt with the highest priority for the low level interrupt line. 0 No pending interrupt. 1h Interrupt pending for channel 0, error Group1. ... 20h Interrupt pending for channel 31, error Group1. 21h Interrupt pending for channel 32, error Group1. ... 60h Interrupt pending for channel 63, error Group1. Note: Reading the interrupt vector will not clear the corresponding flag in the ESMSR1 register. Group2 interrupts are fixed to the high level interrupt line only. User and privileged mode (write): Writes have no effect.

28.2.4.13 ESMLTCR Register (Offset = 30h) [reset = 0h]

ESMLTCR is shown in Figure 28-40 and described in Table 28-29.

Return to Summary Table.

ESM Low-Time Counter Register

Figure 28-40. ESMLTCR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LTCP															
R/W-0h																R/W-0h															

Table 28-29. ESMLTCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
15-0	LTCP	R/W	0h	ERROR Pin Low-Time Counter 16bit pre-loadable down-counter to control low-time of ERROR pin. The low-time counter is triggered by the peripheral clock (VCLK). Note: Low time counter is set to the default preload value of the ESMLTCPR in the following cases: 1. Reset (power on reset or warm reset) 2. An error occurs 3. User forces an error

28.2.4.14 ESMLTCPR Register (Offset = 34h) [reset = 0h]

ESMLTCPR is shown in Figure 28-41 and described in Table 28-30.

Return to Summary Table.

ESM Low-Time Counter Preload Register

Figure 28-41. ESMLTCPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LTCP															
R/W-0h																R/W-0h															

Table 28-30. ESMLTCPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
15-0	LTCP	R/W	0h	ERROR Pin Low-Time Counter Pre-load Value 16bit pre-load value for the ERROR pin low-time counter. Note: Only LTCP.15 and LTCP.14 are configurable (privileged mode write).

28.2.4.15 ESMEKR Register (Offset = 38h) [reset = 0h]

ESMEKR is shown in Figure 28-42 and described in Table 28-31.

Return to Summary Table.

ESM Error Key Register

Figure 28-42. ESMEKR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EKEY			
R/W-0h																												R/W-0h			

Table 28-31. ESMEKR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
3-0	EKEY	R/W	0h	Error Key. The key to reset the ERROR pin or to force an error on the ERROR pin. User and privileged mode (read): Returns current value of the EKEY. Privileged mode (write): 0 Activates normal mode (recommended default mode). Ah Forces error on ERROR pin. 5h The ERROR pin set to high when the low time counter (LTC) has completed; then the EKEY bit will switch back to normal mode (EKEY = 0000) All other values Activates normal mode.

28.2.4.16 ESMSSR2 Register (Offset = 3Ch) [reset = 0h]

ESMSSR2 is shown in Figure 28-43 and described in Table 28-32.

Return to Summary Table.

ESM Status Shadow Register 2

Figure 28-43. ESMSSR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

Table 28-32. ESMSSR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Shadow register for status information on pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred. Write: Leaves the bit unchanged. 1 Read: Error occurred. Write: Clears the bit. ESMSR2 is not impacted by this action. Note: Errors are stored until they are cleared by the software or at power-on reset (PORRST).

28.2.4.17 ESMIEPSR4 Register (Offset = 40h) [reset = 0h]

ESMIEPSR4 is shown in Figure 28-44 and described in Table 28-33.

Return to Summary Table.

ESM Enable ERROR Pin Action/Response Register 4

Figure 28-44. ESMIEPSR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPSET																															
R/W-0h																															

Table 28-33. ESMIEPSR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IEPSET	R/W	0h	Enable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding clear bit in the ESMIEPCR4 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Enables failure influence on ERROR pin and sets the corresponding clear bit in the ESMIEPCR4 register.

28.2.4.18 ESMIEPCR4 Register (Offset = 44h) [reset = 0h]

ESMIEPCR4 is shown in Figure 28-45 and described in Table 28-34.

Return to Summary Table.

ESM Disable ERROR Pin Action/Response Register 4

Figure 28-45. ESMIEPCR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPCLR																															
R/W-0h																															

Table 28-34. ESMIEPCR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IEPCLR	R/W	0h	Disable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding set bit in the ESMIEPSR4 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Disables failure influence on ERROR pin and clears the corresponding set bit in the ESMIEPSR4 register.

28.2.4.19 ESMIESR4 Register (Offset = 48h) [reset = 0h]

ESMIESR4 is shown in Figure 28-46 and described in Table 28-35.

Return to Summary Table.

ESM Interrupt Enable Set/Status Register 4

Figure 28-46. ESMIESR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSET																															
R/W-0h																															

Table 28-35. ESMIESR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTENSET	R/W	0h	Set interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding clear bit in the ESMIECR4 register unchanged. 1 Read: Interrupt is enabled. Write: Enables interrupt and sets the corresponding clear bit in the ESMIECR4 register.

28.2.4.20 ESMIECR4 Register (Offset = 4Ch) [reset = 0h]

ESMIECR4 is shown in Figure 28-47 and described in Table 28-36.

Return to Summary Table.

ESM Interrupt Enable Clear/Status Register 4

Figure 28-47. ESMIECR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENCLR																															
R/W-0h																															

Table 28-36. ESMIECR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTENCLR	R/W	0h	Clear Interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding set bit in the ESMIESR4 register unchanged. 1 Read: Interrupt is enabled. Write: Disables interrupt and clears the corresponding set bit in the ESMIESR4 register.

28.2.4.21 ESMILSR4 Register (Offset = 50h) [reset = 0h]

ESMILSR4 is shown in Figure 28-48 and described in Table 28-37.

Return to Summary Table.

Interrupt Level Set/Status Register 4

Figure 28-48. ESMILSR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLSET																															
R/W-0h																															

Table 28-37. ESMILSR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLVLSET	R/W	0h	Set Interrupt Priority Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding clear bit in the ESMILCR4 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to high level interrupt line and sets the corresponding clear bit in the ESMILCR4 register.

28.2.4.22 ESMILCR4 Register (Offset = 54h) [reset = 0h]

ESMILCR4 is shown in Figure 28-49 and described in Table 28-38.

Return to Summary Table.

Interrupt Level Clear/Status Register 4

Figure 28-49. ESMILCR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLCLR																															
R/W-0h																															

Table 28-38. ESMILCR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLVLCLR	R/W	0h	Clear Interrupt Priority. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding set bit in the ESMILSR4 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to low level interrupt line and clears the corresponding set bit in the ESMILSR4 register.

28.2.4.23 ESMSR4 Register (Offset = 58h) [reset = 0h]

ESMSR4 is shown in Figure 28-50 and described in Table 28-39.

Return to Summary Table.

ESM Status Register 4

Figure 28-50. ESMSR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

Table 28-39. ESMSR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. Note: After nRST, if one of these flags are set and the corresponding interrupt are enabled, the interrupt service routine will be called.

28.2.4.24 ESMIEPSR7 Register (Offset = 80h) [reset = 0h]

ESMIEPSR7 is shown in Figure 28-51 and described in Table 28-40.

Return to Summary Table.

ESM Enable ERROR Pin Action/Response Register 7

Figure 28-51. ESMIEPSR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPSET																															
R/W-0h																															

Table 28-40. ESMIEPSR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IEPSET	R/W	0h	Enable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding clear bit in the ESMIEPCR7 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Enables failure influence on ERROR pin and sets the corresponding clear bit in the ESMIEPCR7 register.

28.2.4.25 ESMIEPCR7 Register (Offset = 84h) [reset = 0h]

ESMIEPCR7 is shown in Figure 28-52 and described in Table 28-41.

Return to Summary Table.

ESM Disable ERROR Pin Action/Response Register 7

Figure 28-52. ESMIEPCR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPCLR																															
R/W-0h																															

Table 28-41. ESMIEPCR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IEPCLR	R/W	0h	Disable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding set bit in the ESMIEPSR7 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Disables failure influence on ERROR pin and clears the corresponding set bit in the ESMIEPSR7 register.

28.2.4.26 ESMIESR7 Register (Offset = 88h) [reset = 0h]

ESMIESR7 is shown in Figure 28-53 and described in Table 28-42.

Return to Summary Table.

ESM Interrupt Enable Set/Status Register 7

Figure 28-53. ESMIESR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSET																															
R/W-0h																															

Table 28-42. ESMIESR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTENSET	R/W	0h	Set interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding clear bit in the ESMIECR7 register unchanged. 1 Read: Interrupt is enabled. Write: Enables interrupt and sets the corresponding clear bit in the ESMIECR7 register.

28.2.4.27 ESMIECR7 Register (Offset = 8Ch) [reset = 0h]

ESMIECR7 is shown in Figure 28-54 and described in Table 28-43.

Return to Summary Table.

ESM Interrupt Enable Clear/Status Register 7

Figure 28-54. ESMIECR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENCLR																															
R/W-0h																															

Table 28-43. ESMIECR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTENCLR	R/W	0h	Clear Interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding set bit in the ESMIESR7 register unchanged. 1 Read: Interrupt is enabled. Write: Disables interrupt and clears the corresponding set bit in the ESMIESR7 register.

28.2.4.28 ESMILSR7 Register (Offset = 90h) [reset = 0h]

ESMILSR7 is shown in Figure 28-55 and described in Table 28-44.

Return to Summary Table.

Interrupt Level Set/Status Register 7

Figure 28-55. ESMILSR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLSET																															
R/W-0h																															

Table 28-44. ESMILSR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLVLSET	R/W	0h	Set Interrupt Priority Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding clear bit in the ESMILCR7 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to high level interrupt line and sets the corresponding clear bit in the ESMILCR7 register.

28.2.4.29 ESMILCR7 Register (Offset = 94h) [reset = 0h]

ESMILCR7 is shown in Figure 28-56 and described in Table 28-45.

Return to Summary Table.

Interrupt Level Clear/Status Register 7

Figure 28-56. ESMILCR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLCLR																															
R/W-0h																															

Table 28-45. ESMILCR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTLVLCLR	R/W	0h	Clear Interrupt Priority. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding set bit in the ESMILSR7 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to low level interrupt line and clears the corresponding set bit in the ESMILSR7 register.

28.2.4.30 ESMSR7 Register (Offset = 98h) [reset = 0h]

ESMSR7 is shown in Figure 28-57 and described in Table 28-46.

Return to Summary Table.

ESM Status Register 7

Figure 28-57. ESMSR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

Table 28-46. ESMSR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. Note: After nRST, if one of these flags are set and the corresponding interrupt are enabled, the interrupt service routine will be called.

28.3 Cyclic Redundancy Check (CRC)

This section describes the cyclic redundancy check (CRC) controller module.

NOTE: This section describes a superset implementation of the CRC module that includes features and functionality that require DMA. Because not all devices have DMA capability, consult the device-specific datasheet to determine applicability of these features and functions to the device being used.

28.3.1 Overview

The CRC controller is a module that is used to perform CRC (Cyclic Redundancy Check) to verify the integrity of memory system. A signature representing the contents of the memory is obtained when the contents of the memory are read into CRC controller. The responsibility of CRC controller is to calculate the signature for a set of data and then compare the calculated signature value against a pre-determined good signature value. CRC controller supports two channels to perform CRC calculation on multiple memories in parallel and can be used on any memory system.

28.3.1.1 Features

The CRC controller offers:

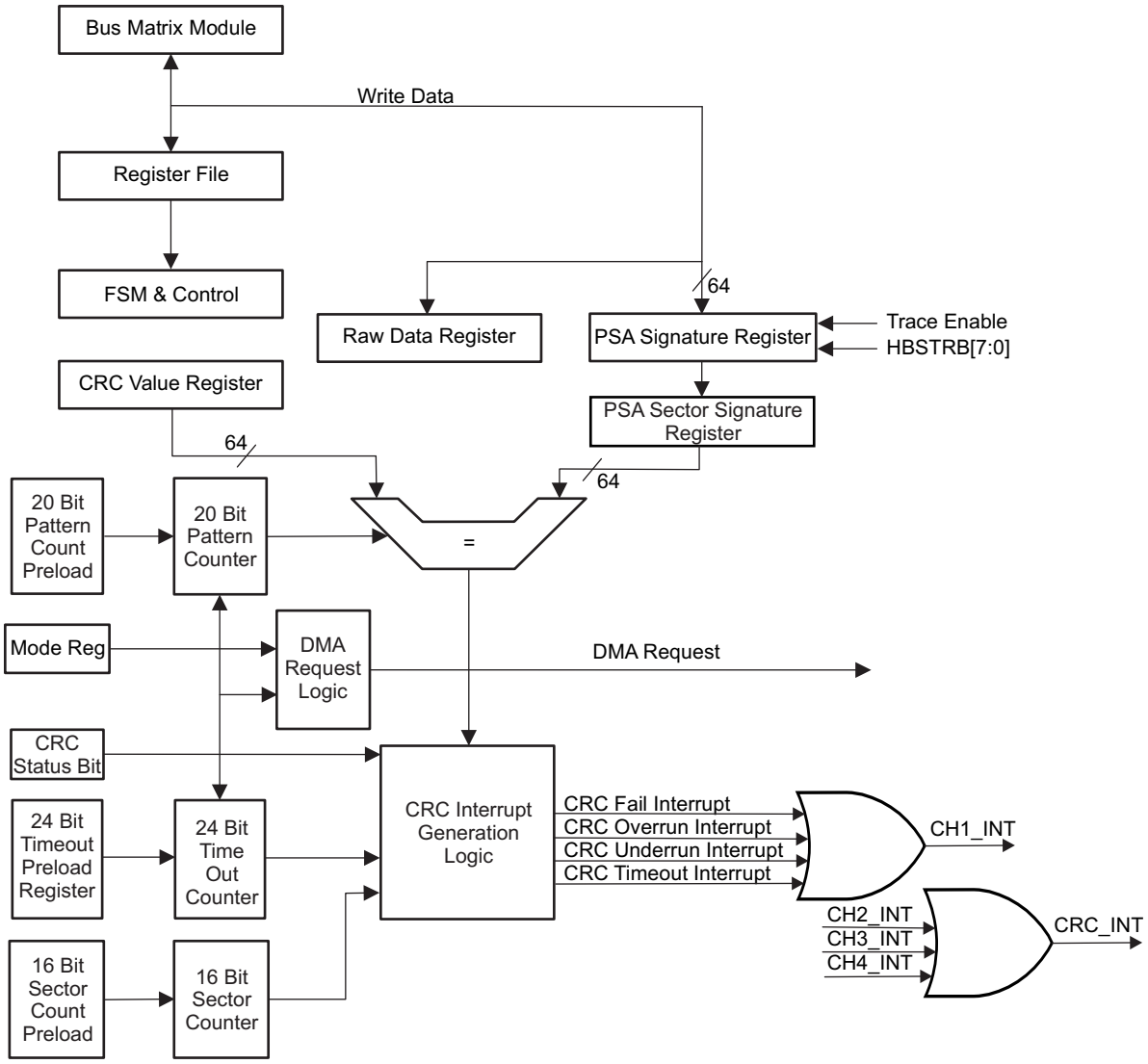
- Two channels to perform background signature verification on any memory sub-system.
- Data compression on 8, 16, 32, and 64 bit data size.
- Maximum-length PSA (Parallel Signature Analysis) register constructed based on 64 bit primitive polynomial.
- Each channel has a CRC Value Register that contains the pre-determined CRC value.
- Use timed base event trigger from timer to initiate DMA data transfer.
- Programmable 20-bit pattern counter per channel to count the number of data patterns for compression.
- Three modes of operation. Auto, Semi-CPU and Full-CPU.
- For each channel, CRC can be performed either by CRC Controller or by CPU.
- Automatically perform signature verification without CPU intervention in AUTO mode.
- Generate interrupt to CPU in Semi-CPU mode to allow CPU to perform signature verification itself.

- Generate CRC fail interrupt in AUTO mode if signature verification fails.
- Generate Timeout interrupt if CRC is not performed within the time limit.
- Generate DMA request per channel to initiate CRC value transfer.

28.3.1.2 Block Diagram

Figure 28-58 shows a block diagram of the CRC controller.

Figure 28-58. CRC Controller Block Diagram For One Channel



28.3.2 Module Operation

28.3.2.1 General Operation

There are two channels in CRC controller and for each channel there is a memory mapped PSA (Parallel Signature Analysis) Signature Register and a memory mapped CRC (Cyclic Redundancy Check) Value register. A memory can be organized into multiple sectors with each sector consisting of multiple data patterns. A data pattern can be 8-, 16-, 32-, or 64-bit data. CRC module performs the signature calculation and compares the signature to a pre-determined value. The PSA Signature Register compresses an incoming data pattern into a signature when it is written. When one sector of data patterns are written into PSA Signature Register, a final signature corresponding to the sector is obtained. CRC Value Register stores the pre-determined signature corresponding to one sector of data patterns. The calculated signature and the pre-determined signature are then compared to each other for signature verification. To minimize CPU's involvement, data patterns transfer can be carried out at the background of CPU using DMA controller. DMA is setup to transfer data from memory from which the contents to be verified to the memory mapped PSA Signature Register. When DMA transfers data to the memory mapped PSA Signature Register, a signature is generated. A programmable 20-bit data pattern counter is used for each channel to define the number of data patterns to calculate for each sector. Signature verification can be performed automatically by CRC controller in AUTO mode or by CPU itself in Semi-CPU or Full-CPU mode. In AUTO mode, a self sustained CRC signature calculation can be achieved without any CPU intervention.

28.3.2.2 CRC Modes of Operation

CRC Controller can operate in AUTO, Semi-CPU, and Full-CPU modes.

28.3.2.2.1 AUTO Mode

In AUTO mode, CRC Controller in conjunction with DMA controller can perform CRC totally without CPU intervention. A sustained transfer of data to both the PSA Signature Register and CRC Value Register are performed in the background of CPU. When a mismatch is detected, an interrupt is generated to CPU. A 16 bit current sector ID register is provided to identify which sector causes a CRC failure.

28.3.2.2.2 Semi-CPU Mode

In Semi-CPU mode, DMA controller is also utilized to perform data patterns transfer to PSA Signature Register. Instead of performing signature verification automatically, the CRC controller generates an compression complete interrupt to CPU after each sector is compressed. Upon responding to the interrupt the CPU performs the signature verification by reading the calculated signature stored at the PSA Sector Signature Register and compare it to a pre-determined CRC value.

28.3.2.2.3 Full CPU Mode

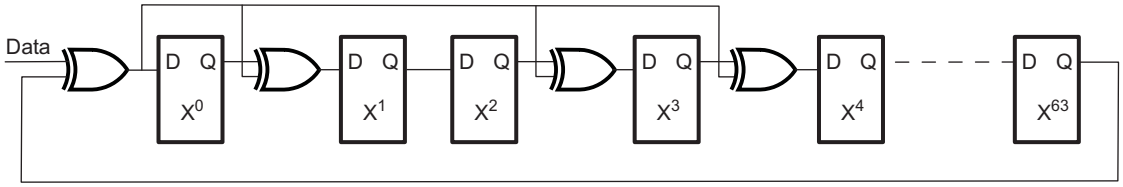
In Full-CPU mode, the CPU does the data patterns transfer and signature verification all by itself. When CPU has enough throughput, it can perform data patterns transfer by reading data from the memory system to the PSA Signature Register. After certain number of data patterns are compressed, the CPU can read from the PSA Signature Register and compare the calculated signature to the pre-determined CRC signature value. In Full-CPU mode, neither interrupt nor DMA request is generated. All counters are also disabled.

28.3.2.3 PSA Signature Register

The 64-bit PSA Signature Register is based on the primitive polynomial (as in the following equation) to produce the maximum length LFSR (Linear Feedback Shift Register), as shown in Figure 28-59.

$$f(x) = x^{64} + x^4 + x^3 + x + 1 \quad (22)$$

Figure 28-59. Linear Feedback Shift Register (LFSR)



The serial implementation of LFSF has a limitation that, it requires ‘n’ clock cycles to calculate the CRC values for an ‘n’ bit data stream. The idea is to produce the same CRC value operating on a multi-bit data stream, as would occur if the CRC were computed one bit at a time over the whole data stream. The algorithm involves looping to simulate the shifting, and concatenating strings to build the equations after ‘n’ shift.

The parallel CRC calculation based on the polynomial can be illustrated in the following HDL code:

```
for i in 63 to 0 loop
   NEXT_CRC_VAL(0) := CRC_VAL(63) xor DATA(i);
     for j in 1 to 63 loop
       case j is
          when 1|3|4 =>
               NEXT_CRC_VAL(j) :=
                   CRC_VAL(j - 1) xor CRC_VAL(63) xor DATA(i);
          when others =>
               NEXT_CRC_VAL(j) := CRC_VAL(j - 1);
       end case;
     end loop;
   CRC_VAL := NEXT_CRC_VAL;
end loop;
```

NOTE: 1) The inner loop is to calculate the next value of each shift register bit after one cycle

2) The outer loop is to simulate 64 cycles of shifting. The equation for each shift register bit is thus built before it is compressed into the shift register.

3) MSB of the DATA is shifted in first

There is one PSA Signature Register per CRC channel. PSA Signature Register can be both read and written. When it is written, it can either compress the data or just capture the data depending on the state of CHx_MODE bits. If CHx_MODE=Data Capture, a seed value can be planted in the PSA Signature Register without compression. Other modes other than Data Capture will result with the data compressed by PSA Signature Register when it is written. Each channel can be planted with different seed value before compression starts. When PSA Signature Register is read, it gives the calculated signature.

CRC Controller should be used in conjunction with the on chip DMA controller to produce optimal system performance. The incoming data pattern to PSA Signature Register is typically initiated by the DMA master. When DMA is properly setup, it would read data from the pre-determined memory system and write them to the memory mapped PSA Signature Register. Each time PSA Signature Register is written a signature is generated. CPU itself can also perform data transfer by reading from the memory system and perform write operation to PSA Signature Register if CPU has enough throughput to handle data patterns transfer.

After system reset and when AUTO mode is enabled, CRC Controller automatically generates a DMA request to request the pre-determined CRC value corresponding to the first sector of memory to be checked.

In AUTO mode, when one sector of data patterns is compressed, the signature stored at the PSA Signature Register is first copied to the PSA Sector Signature Register and PSA Signature Register is then cleared out to all zeros. An automatic signature verification is then performed by comparing the signature stored at the PSA Sector Signature Register to the CRC Value Register. After the comparison the CRC Controller can generate a DMA request. Upon receiving the DMA request the DMA controller will update the CRC Value Register by transferring the next pre-determined signature value associated with the next sector of memory system. If the signature verification fails then CRC Controller can generate a CRC fail interrupt.

In Full-CPU mode, no DMA request and interrupt are generated at all. The number of data patterns to be compressed is determined by CPU itself. Full-CPU mode is useful when DMA controller is not available to perform background data patterns transfer. The OS can periodically generate a software interrupt to CPU and use CPU to accomplish data transfer and signature verification.

CRC Controller supports doubleword, word, half word and byte access to the PSA Signature Register. During a non-doubleword write access, all unwritten byte lanes are padded with zero's before compression. Note that comparison between PSA Sector Signature Register and CRC Value Register is always in 64 bit because a compressed value is always expressed in 64 bit.

There is a software reset per channel for PSA Signature Register. When set, the PSA Signature Register is reset to all zeros.

PSA Signature Register is reset to zero under the following conditions:

- System reset
- PSA Software reset
- One sector of data patterns are compressed

28.3.2.4 PSA Sector Signature Register

After one sector of data is compressed, the final resulting signature calculated by PSA Signature Register is transferred to the PSA Sector Signature Register. PSA Signature Register is a read only register. During Semi-CPU mode, the host CPU should read from the PSA Sector Signature Register instead of reading from PSA Signature Register for signature verification to avoid data coherency issue. The PSA Signature Register can be updated with new signature before the host CPU is able to retrieve it.

In Semi-CPU mode, no DMA request is generated. When one sector of data patterns is compressed, CRC controller first generates a compression complete interrupt. Responding to the interrupt, CPU will in the ISR read the PSA Sector Signature Register and compare it to the known good signature or write the signature value to another memory location to build a signature file. In Semi-CPU mode, CPU must perform the signature verification in a manner to prevent any overrun condition. The overrun condition occurs when the compression complete interrupt is generated after one sector of data patterns is compressed and CPU has not read from the PSA Sector Signature Register to perform necessary signature verification before PSA Sector Signature Register is overridden with a new value. An overrun interrupt can be enable to generate when overrun condition occurs. During Semi-CPU mode, the host CPU should read from the PSA Sector Signature Register instead of reading from PSA Signature Register for signature verification to avoid data coherency issue. The PSA Signature Register can be updated with new signature before the host CPU is able to retrieve it.

28.3.2.5 CRC Value Register

Associated with each channel there is a CRC Value Register. The CRC Value Register stores the pre-determined CRC value. After one sector of data patterns is compressed by PSA Signature Register, CRC Controller can automatically compare the resulting signature stored at the PSA Sector Signature Register with the pre-determined value stored at the CRC Value Register if AUTO mode is enabled. If the signature verification fails, CRC Controller can be enabled to generate an CRC fail interrupt. When the channel is set up for Semi-CPU mode, CRC controller first generates a compression complete interrupt to CPU. Upon servicing the interrupt, CPU will then read the PSA Sector Signature Register and then read the corresponding CRC value stored at another location and compare them. CPU should not read from the CRC Value Register during Semi-CPU or Full-CPU mode because the CRC Value Register is not updated during these two modes.

In AUTO mode, for first sector's signature, DMA request is generated when mode is programmed to AUTO. For subsequent sectors, DMA request is generated after each sector is compressed. Responding to the DMA request, DMA controller reloads the CRC Value Register for the next sector of memory system to be checked.

When CRC Value Register is updated with a new CRC value, an internal flag is set to indicate that CRC Value Register contains the most current value. This flag is cleared when CRC comparison is performed. Each time at the end of the final data pattern compression of a sector, CRC Controller first checks to see if the corresponding CRC Value Register has the most current CRC value stored in it by polling the flag. If the flag is set then the CRC comparison can be performed. If the flag is not set then it means the CRC Value Register contains stale information. A CRC underrun interrupt is generated. When an underrun condition is detected, signature verification is not performed.

CRC Controller supports doubleword, word, half word and byte access to the CRC Value Register. As noted before comparison between PSA Sector Signature Register and CRC Value Register during AUTO mode is carried out in 64 bit.

28.3.2.6 Raw Data Register

The raw or un-compressed data written to the PSA Signature Register is also saved in the Raw Data Register. This register is read only.

28.3.2.7 Example DMA Controller Setup

DMA controller needs to be setup properly in either either AUTO or Semi-CPU mode as DMA controller is used to transfer data patterns. Hardware or a combination of hardware and software DMA triggering are supported.

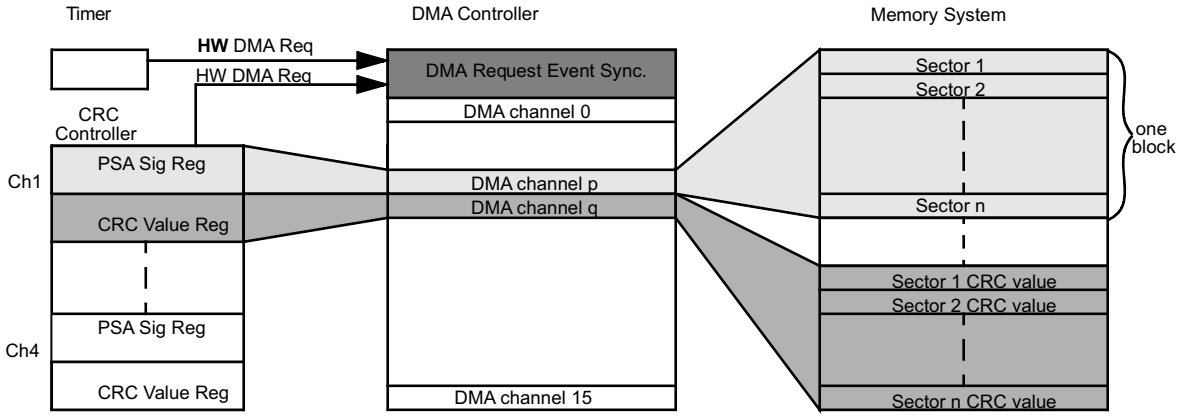
28.3.2.7.1 AUTO Mode Using Hardware Timer Trigger

There are two DMA channels associated with each CRC channel when in AUTO mode. One DMA channel is setup to transfer data patterns from the source memory to the PSA Signature Register. The second DMA channel is setup to transfer the pre-determined signature to the CRC Value Register. The trigger source for the first DMA channel can be either by hardware or by software. As illustrated in Figure 28-60 a timer can be used to trigger a DMA request to initiate transfer from the source memory system to PSA Signature Register. In AUTO mode, CRC Controller also generates DMA request after one sector of data patterns is compressed to initiate transfer of the next CRC value corresponding to the next sector of memory. Thus a new CRC value is always updated in the CRC Value Register by DMA synchronized to each sector of memory.

A block of memory system is usually divided into many sectors. All sectors are the same size. The sector size is programmed in the CRC_PCOUNT_REGx and the number of sectors in one block is programmed in the CRC_SCOUNT_REGx of the respective channel. CRC_PCOUNT_REGx multiplies CRC_SCOUNT_REGx and multiplies transfer size of each data pattern should give the total block size in number of bytes.

The total size of the memory system to be examined is also programmed in the respective transfer count register inside DMA module. The DMA transfer count register is divided into two parts. They are element count and frame count. Note that an HW DMA request can be programmed to trigger either one frame or one entire block transfer. In Figure 28-60, an HW DMA request from a timer is used as a trigger source to initiate DMA transfer. If all four CRC channels are active in AUTO mode then a total of four DMA requests would be generated by CRC Controller.

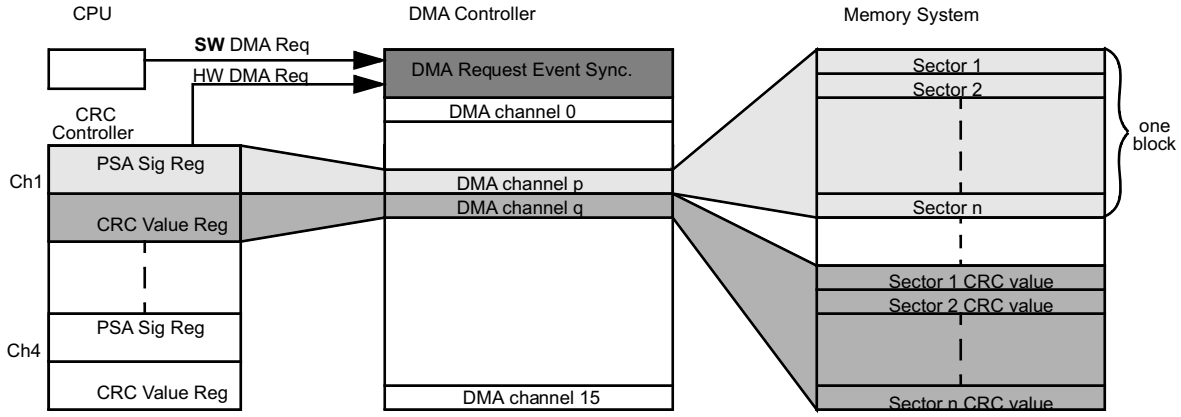
Figure 28-60. AUTO Mode Using Hardware Timer Trigger



28.3.2.7.2 AUTO Mode Using Software Trigger

The data patterns transfer can also be initiated by software. CPU can generate a software DMA request to activate the DMA channel to transfer data patterns from source memory system to the PSA Signature Register. To generate a software DMA request CPU needs to set the corresponding DMA channel in the DMA software trigger register. Note that just one software DMA request from CPU is enough to complete the entire data patterns transfer for all sectors. See Figure 28-61 for an illustration.

Figure 28-61. AUTO Mode With Software CPU Trigger



28.3.2.7.3 Semi-CPU Mode Using Hardware Timer Trigger

During semi-CPU mode, no DMA request is generated by CRC controller. Therefore, no DMA channel is allocated to update CRC Value Register. CPU should not read from CRC Value Register in semi-CPU mode as it contains stale value. Note that no signature verification is performed at all during this mode. Similar to AUTO mode, either by hardware or by software DMA request can be used as a trigger for data patterns transfer. Figure 28-62 illustrates the DMA setup using semi-CPU mode with hardware timer trigger.

Figure 28-62. Semi-CPU Mode With Hardware Timer Trigger

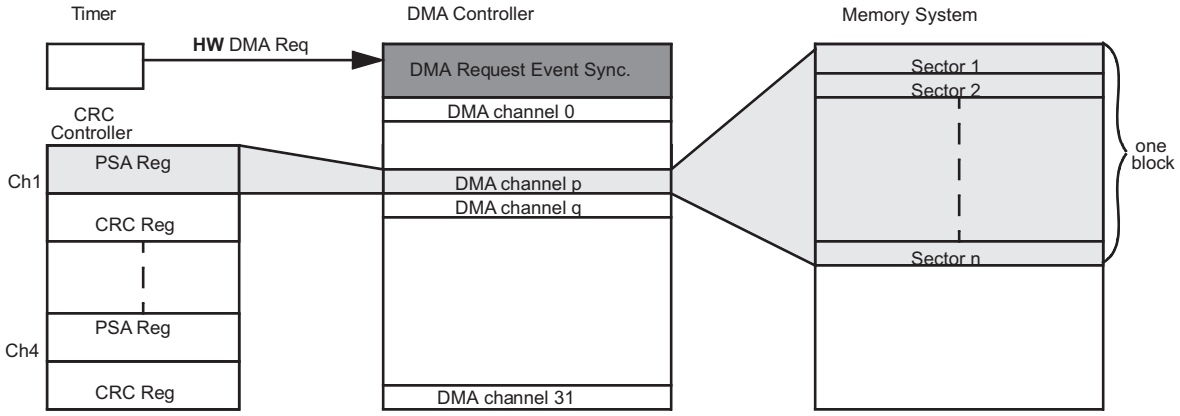


Table 28-47. CRC Modes in Which DMA Request and Counter Logic are Active or Inactive

Mode	DMA Request	Pattern Counter	Sector Counter	Timeout Counter
AUTO	Active	Active	Active	Active
Semi-CPU	Inactive	Active	Active	Active
Full-CPU	Inactive	Inactive	Inactive	Inactive

28.3.2.8 Pattern Count Register

There is a 20-bit data pattern counter for every CRC channel. The data pattern counter is a down counter and can be pre-loaded with a programmable value stored in the Pattern Count Register. When the data pattern counter reaches zero, a compression complete interrupt is generated in Semi-CPU mode and an automatic signature verification is performed in AUTO mode. In AUTO only, DMA request is generated to trigger the DMA controller to update the CRC Value Register.

NOTE: The data pattern count should be divisible by the total transfer count as programmed in DMA controller. The total transfer count is the product of element count and frame count.

28.3.2.9 Sector Count Register/Current Sector Register

Each channel contains a 16 bit sector counter. The sector count register stores the number of sectors. Sector counter is a free running counter and is incremented by one each time when one sector of data patterns is compressed. When the signature verification fails, the current value stored in the sector counter is saved into current sector register. If signature verification fails, CPU can read from the current sector register to identify the sector which causes the CRC mismatch. To aid and facilitate the CPU in determining the cause of a CRC failure, it is advisable to use the following equation during CRC and DMA setup:

CRC Pattern Count × CRC Sector Count = DMA Element Count × DMA Frame Count

The current sector register is frozen from being updated until both the current sector register is read and CRC fail status bit is cleared by CPU. If CPU does not respond to the CRC failure in a timely manner before another sector produces a signature verification failure, the current sector register is not updated with the new sector number. An overrun interrupt is generate instead. If current sector register is already frozen with an erroneous sector and emulation is entered with SUSPEND signal goes to high then the register still remains frozen even it is read.

In Semi-CPU mode, the current sector register is used to indicate the sector for which the compression complete has last happened.

The current sector register is reset when the PSA software reset is enabled.

NOTE: Both data pattern count and sector count registers must be greater than or equal to one for the counters to count. After reset, pattern count and sector count registers default to zero and the associated counters are inactive.

28.3.2.10 Interrupt

The CRC controller generates several types of interrupts per channel. Associated with each interrupt, there is an interrupt enable bit. No interrupt is generated in Full-CPU mode.

- Compression complete interrupt
- CRC fail interrupt
- Overrun interrupt
- Underrun interrupt
- Timeout interrupt

Table 28-48. Modes in Which Interrupt Condition Can Occur

	AUTO	Semi-CPU	Full-CPU
Compression Complete	no	yes	no
CRC Fail	yes	no	no
Overrun	yes	yes	no
Underrun	yes	no	no
Timeout	yes	yes	no

28.3.2.10.1 Compression Complete Interrupt

Compression complete interrupt is generated in Semi-CPU mode only. When the data pattern counter reaches zero, the compression complete flag is set and the interrupt is generated.

28.3.2.10.2 CRC Fail Interrupt

CRC fail interrupt is generated in AUTO mode only. When the signature verification fails, the CRC fail flag is set,. CPU should take action to address the fail condition and clear the CRC fail flag after it resolves the CRC mismatch.

28.3.2.10.3 Overrun Interrupt

Overrun interrupt is generated in either AUTO or Semi-CPU mode. During AUTO mode, if a CRC fail is detected then the current sector number is recorded in the current sector register. If CRC fail status bit is not cleared and current sector register is not read by the host CPU before another CRC fail is detected for another sector then an overrun interrupt is generated. During Semi-CPU mode, when the data pattern counter finishes counting, it generates a compression complete interrupt. At the same time the signature is copied into the PSA Sector Signature Register. If the host CPU does not read the signature from PSA Sector Signature Register before it is updated again with a new signature value then an overrun interrupt is generated.

28.3.2.10.4 Underrun Interrupt

Underrun interrupt only occurs in AUTO mode. The interrupt is generated when the CRC Value Register is not updated with the corresponding signature when the data pattern counter finishes counting. During AUTO mode, CRC Controller generates DMA request to update CRC Value Register in synchronization to the corresponding sector of the memory. Signature verification is also performed if underrun condition is detected. And CRC fail interrupt is generated at the same time as the underrun interrupt.

28.3.2.10.5 Timeout Interrupt

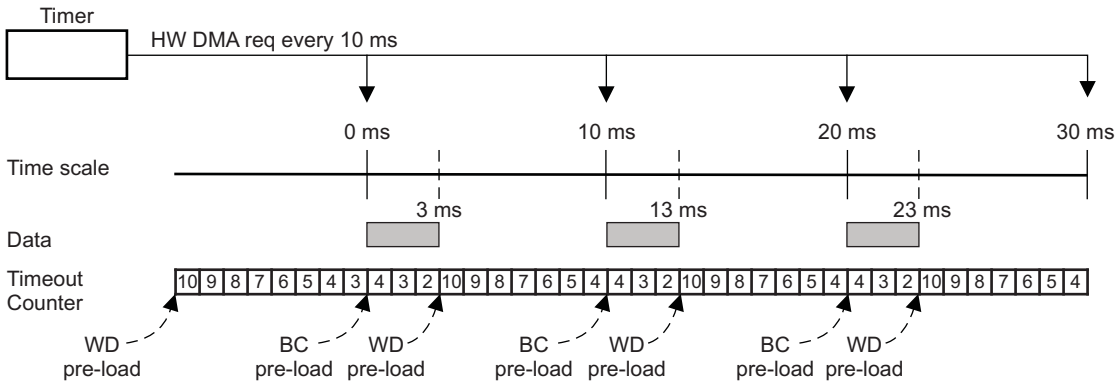
To ensure that the memory system is examined within a pre-defined time frame and no loss of incoming data there is a 24 bit timeout counter per CRC channel. The 24 bit timeout down counter can be pre-loaded with two different pre-load values, watchdog timeout pre-load value (CRC_WDTOPLDx) and block complete timeout pre-load value (CRC_BCTOPLDx). The timeout counter is clocked by a prescaler clock which is permanently running at division 64 of HCLK clock.

First pattern of data must be transferred by the DMA before the timeout counter expires, Watchdog timeout pre-load register (CRC_WDTOPLDx) is used as timeout counter. Block complete timeout pre-load register (CRC_BCTOPLDx) is used to check if one complete block of data patterns are compressed within a specific time frame. The timeout counter is first pre-loaded with CRC_WDTOPLDx after either AUTO or Semi-CPU mode is selected and starts to down count. If the timeout counter expires before DMA transfers any data pattern to PSA Signature Register then a timeout interrupt is generated. An incoming data pattern before the timeout counter expires will automatically pre-load the timeout counter with CRC_BCTOPLDx the block complete timeout pre-load value.

Block complete timeout pre-load value is used to check it one block of data patterns are compressed within a given time limit. If the timeout counter pre-loaded with CRC_BCTOPLDx value expires before one block of data patterns are compressed a timeout interrupt is generated. When one block (pattern count x sector count) of data patterns are compressed before the counter has expired, the counter is pre-loaded with CRC_WDTOPLDx value again. If the timeout counter is pre-loaded with zero then the counter is disable and no timeout interrupt is generated.

In Figure 28-63, a timer generates DMA request every 10ms to trigger one block (pattern count x sector count) transfer. Since we want to make sure that DMA does start to transfer a block every 10 ms we would set the first pre-load value to 10ms in CRC_WDTOPLDx. We also want to make sure that one block of data patterns are compressed within 4ms. With such a requirement, we would set the second pre-load value to 4ms in CRC_BCTOPLDx register.

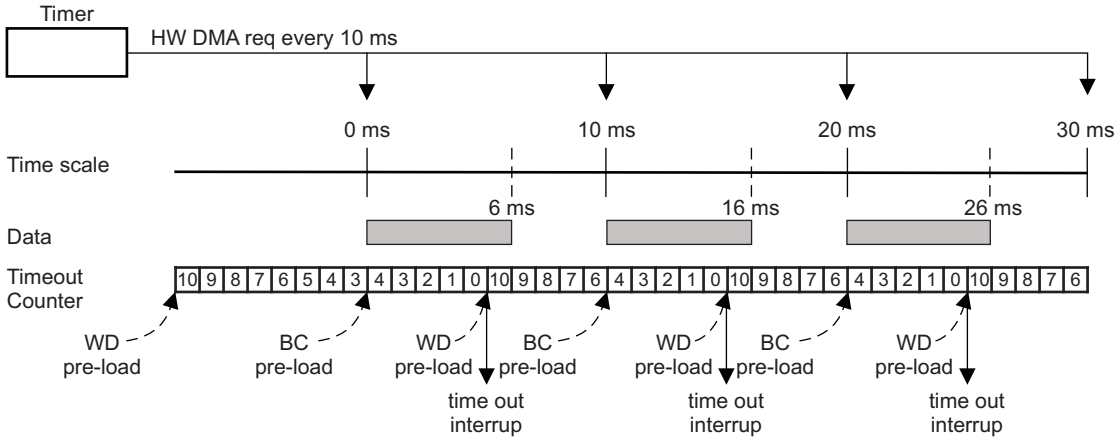
Figure 28-63. Timeout Example 1



WD pre-load = watchdog timeout pre-load (CRC_WDTOPLDx)
BC pre-load = block complete timeout pre-load (CRC_BCTOPLDx)

Note: No timeout interrupt is generated in this example since each block of data patterns are compressed in 3 ms and DMA does initiate a block transfer every 10 ms.

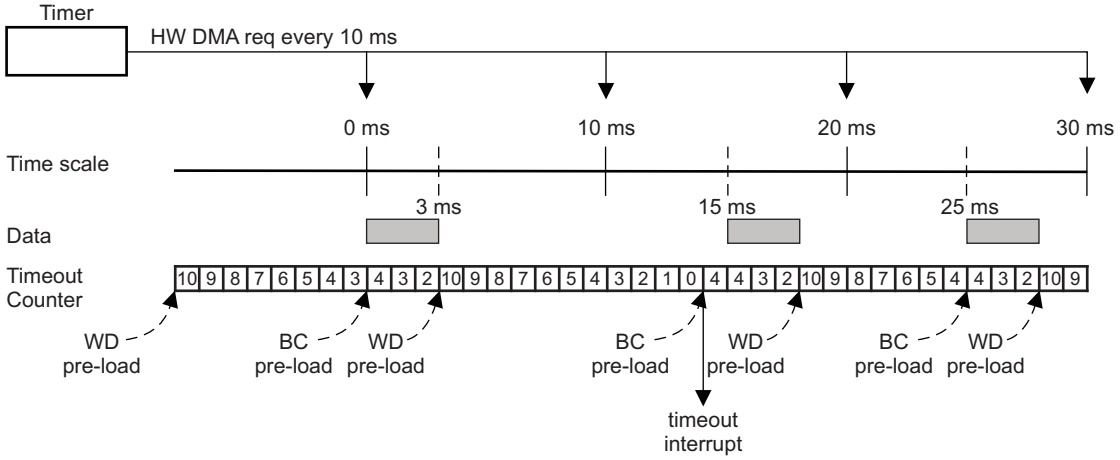
Figure 28-64. Timeout Example 2



WD pre-load = watchdog timeout pre-load (CRC_WDTOPLDx)
BC pre-load = block complete timeout pre-load (CRC_BCTOPLDx)

Note: Timeout interrupt is generated in this example since each block of data patterns are compressed in 6 ms and this is out of the 4ms time frame.

Figure 28-65. Timeout Example 3



WD pre-load = watchdog timeout pre-load (CRC_WDTOPLDx)
BC pre-load = block complete timeout pre-load (CRC_BCTOPLDx)

Note: Timeout interrupt is generated in this example since DMA can not transfer the second block of data within 10ms time limit and the reason may be that DMA is set up in fixed priority scheme and DMA is serving other higher priority channels at the time before it can service the timer request.

28.3.2.10.6 Interrupt Offset Register

CRC Controller only generates one interrupt request to interrupt manager. A interrupt offset register is provided to indicate the source of the pending interrupt with highest priority. Table 28-49 shows the offset interrupt vector address of each interrupt condition in an ascending order of priority.

Table 28-49. Interrupt Offset Mapping

Offset Value	Interrupt Condition
0	Phantom
1h	Ch1 CRC Fail
2h	Ch2 CRC Fail
3h-8h	Reserved
9h	Ch1 Compression Complete
Ah	Ch2 Compression Complete
Bh-10h	Reserved
11h	Ch1 Overrun
12h	Ch2 Overrun
13h-18h	Reserved
19h	Ch1 Underrun
1Ah	Ch2 Underrun
1Bh-20h	Reserved
21h	Ch1 Timeout
22h	Ch2 Timeout
23h-24h	Reserved

28.3.2.10.7 Error Handling

When an interrupt is generated, host CPU should take appropriate actions to identify the source of error and restart the respective channel in DMA and CRC module. To restart a CRC channel, the user should perform the following steps in the ISR:

1. Write to software reset bit in CRC_CTRL register to reset the respective PSA Signature Register.
2. Reset the CHx_MODE bits to 00 in CRC_CTRL register as Data capture mode.
3. Set the CHx_MODE bits in CRC_CTRL register to desired new mode again.
4. Release software reset.

The host CPU should use byte write to restart each individual channel.

28.3.2.11 Power Down Mode

CRC module can be put into power down mode when the power down control bit PWDN is set. The module wakes up when the PWDN bit is cleared.

28.3.2.12 Emulation

A read access from a register in functional mode can sometimes trigger a certain internal event to follow. For example, reading an interrupt offset register triggers an event to clear the corresponding interrupt status flag. During emulation when SUSPEND signal is high, a read access from any register should only return the register contents to the bus and should not trigger or mask any event as it would have in functional mode. This is to prevent debugger from reading the interrupt offset register during refreshing screen and cause the corresponding interrupt status flag to get cleared. Timeout counters are stopped to generate timeout interrupts in emulation mode. No Peripheral Master bus error should be generated if reading from the unimplemented locations.

28.3.2.13 Peripheral Bus Interface

CRC is a Peripheral slave module. The register interface is similar to other peripheral modules. CRC supports following features:

- Different sizes of burst operation.
- Aligned and unaligned accesses.
- Abort is generated for any illegal address accesses.

28.3.3 Example

This section illustrates several of the ways in which the CRC Controller can be utilized to perform CRC.

28.3.3.1 Example: Auto Mode Using Time Based Event Triggering

A large memory area with 2Mbyte (256k doubleword) is to be checked in the background of CPU. CRC is to be performed every 1K byte (128 doubleword). Therefore there should be 2048 pre-recorded CRC values. For illustration purpose, we map channel 1 CRC Value Register to DMA channel 1 and channel 1 PSA Signature Register to DMA channel 2. Assume all DMA transfers are carried out in 64-bit transfer size.

28.3.3.1.1 DMA Setup

- Set up DMA channel 1 with the starting address from which the pre-determined CRC values are stored. Set up the destination address to the memory mapped channel 1 CRC Value Register. Put the source address at post increment addressing mode and put the destination address at constant addressing mode. Use ***hardware*** DMA request for channel 1 to trigger a ***frame*** transfer.
- Set up DMA channel 2 with the source address from which the contents of memory to be verified. Set up the destination address to the memory mapped channel 1 PSA Signature Register. Program the element transfer count to 128 and the frame transfer count to 2048. Put the source address at post increment addressing mode and put the destination address at constant address mode. Use ***hardware*** DMA request for channel 2 to trigger an entire ***block*** transfer.

28.3.3.1.2 Timer Setup

The timer can be any general purpose timer which is capable of generating a time-based DMA request.

- Set up timer to generate DMA request associated with DMA channel 2. For example, an OS can set up the timer to generate a DMA request every 10ms.

28.3.3.1.3 CRC Setup

- Program the pattern count to 128.
- Program the sector count to 2048.
- For example, we want the entire 2Mbytes to be compressed within 5ms. We can program the block complete timeout pre-load (CRC_BCTOPLDx) value to 15625 (5 ms / (1 HCLK period × 64)) if CRC is operating at 200 MHz.
- Enable AUTO mode and all interrupts.

After AUTO mode is selected, CRC Controller automatically generates a DMA request on channel 1. Around the same time the timer module also generates a DMA request on DMA channel 2. When the first incoming data pattern arrives at the PSA Signature Register, the CRC Controller will compress it. After some time, the DMA controller would update the CRC Value Register with a pre-determined value matching the calculated signature for the first sector of 128 64 bit data patterns. After one sector of data patterns are compressed, the CRC Controller generate a CRC fail interrupt if signature stored at the PSA Sector Signature Register does not match the CRC Value Register. CRC Controller generates a DMA request on DMA channel 1 when one sector of data patterns are compressed. This routine will continue until the entire 2Mbyte are consumed. If the timeout counter reached zero before the entire 2Mbytes are compressed a timeout interrupt is generated. After 2MBytes are transferred, the DMA can generate an interrupt to CPU. The entire operation will continue again when DMA responds to the DMA request from both the timer and CRC Controller. The CRC is performed totally without any CPU intervention.

28.3.3.2 Example: Auto Mode Without Using Time Based Triggering

A small but highly secured memory area with 1kbytes is to be checked in the background of CPU. CRC is to be performed every 1Kbytes. Therefore there is only one pre-recorded CRC value. For illustration purpose, we map channel 1 CRC Value Register to DMA channel 1 and channel 1 PSA Signature Register to DMA channel 2. Assume all transfers carried out by DMA are in 64 bit transfer size.

28.3.3.2.1 DMA Setup

- Set up DMA channel 1 with the source address from which the pre-determined CRC value is stored. Set up the destination address to the memory mapped channel 1 CRC Value Register. Put the source address at constant addressing mode and put the destination address at constant addressing mode. Use ***hardware*** DMA request for channel 1.
- Set up DMA channel 2 with the source address from which the memory area to be verified. Set up the destination address to the memory mapped channel 1 PSA Signature Register. Program the element transfer count to 128 and the frame transfer count to 1. Put the source address at post increment addressing mode and put the destination address at constant address mode. Generate a ***software*** DMA request on channel 2 after CRC has completed its setup. Enable autoinitiation for DMA channel 2.

28.3.3.2.2 CRC Setup

- Program the pattern count to 128.
- Program the sector count to 1.
- Leaving the timeout count register with the reset value of zero means no timeout interrupt is generated.
- Enable AUTO mode and all interrupts.

After AUTO mode is selected, the CRC Controller automatically generates a DMA request on channel 1. At the same time the CPU generates a ***software*** DMA request on DMA channel 2. When the first incoming data pattern arrives at the PSA Signature Register, the CRC Controller will compress it. After some time, the DMA controller would update the CRC Value Register with a pre-determined value matching the calculated signature for the first sector of 128 64 bit data patterns. After one sector of data patterns are compressed, the CRC Controller generates a CRC fail interrupt if signature stored at the PSA Sector Signature Register does not match the CRC Value Register. CRC Controller generates a DMA request on DMA channel 1 again after one sector is compressed. After 1kbytes are transferred, the DMA can generate an interrupt to CPU. Responding to the DMA interrupt CPU can restart the CRC routine by generating a software DMA request onto channel 2 again.

28.3.3.3 Example: Semi-CPU Mode

If DMA controller is available in a system, the CRC module can also operate in semi-CPU mode. This means that CPU can still make use of the DMA to perform data patterns transfer to CRC controller in the background. The difference between semi-CPU mode and AUTO mode is that CRC controller does not automatically perform the signature verification. CRC controllers generates a compression complete interrupt to CPU when the one sector of data patterns are compressed. CPU needs to perform the signature verification itself.

A memory area with 2Mbyte is to be verified with the help of the CPU. CRC operation is to be performed every 1K byte. Since there are 2Mbyte (256k doublewords) of memory to be check and we want to perform a CRC every 1Kbyte (128 doublewords) and therefore there should be 2048 pre-recorded CRC values. In Semi-CPU mode, the CRC Value Register is not updated and contains indeterminate data.

28.3.3.3.1 DMA Setup

Set up DMA channel 1 with the source address from which the memory area to be verified are mapped. Set up the destination address to the memory mapped channel 1 PSA Signature Register. Put the starting address at post increment addressing mode and put the destination address at constant address mode. Use hardware DMA request to trigger an entire block transfer for channel 1. Disable autoinitiation for DMA channel 1.

28.3.3.3.2 Timer Setup

The timer can be any general purpose timer which is capable of generating a time based DMA request.

Set up timer to generate DMA request associated with DMA channel 1. For example, an OS can set up the timer to generate a DMA request every 10ms.

28.3.3.3.3 CRC Setup

- Program the pattern count to 128.
- Program the sector count to 2048.
- For example, we want the entire 2Mbytes to be compressed within 5ms. We can program the block complete timeout pre-load value to 15625 (5 ms / (1 HCLK period × 64)) if CRC is operating at 200 MHz.
- Enable Semi-CPU mode and enable all interrupts.

The timer module first generates a DMA request on DMA channel 1 when it is enabled. When the first incoming data pattern arrives at the PSA Signature Register, the CRC controller will compress it. After one sector of data patterns are compressed, the CRC controller generate a compression complete interrupt. Upon responding to the interrupt the CPU would read from the PSA Sector Signature Register. It is up to the CPU on how to deal with the PSA value just read. It can compare it to a known signature value or it can write it to another memory location to build a signature file or even transfer the signature out of the device via SCI or SPI. This routine will continue until the entire 2Mbyte are consumed. The latency of the interrupt response from CPU can cause overrun condition. If CPU does not read from PSA Sector Signature Register before the PSA value is overridden with the signature of the next sector of memory, an overrun interrupt will be generated by CRC controller.

28.3.3.4 Example: Full-CPU Mode

In a system without the availability of DMA controller, the CRC routine can be operated by CPU provided the CPU has enough throughput. CPU needs to read from the memory area from which CRC is to be performed.

A memory area with 2Mbyte is to be checked with the help of the CPU. CRC verification is to be performed every 1K byte. In CPU mode, the CRC Value Register is not updated and contains indeterminate data.

28.3.3.4.1 CRC Setup

- All control registers can be left in their reset state. Only enable Full-CPU mode.

CPU itself reads from the memory and write the data to the PSA Signature Register inside CRC Controller. When the first incoming data pattern arrives at the PSA Signature Register, the CRC Controller will compress it. After ***2MBytes*** data patterns are compressed, CPU can read from the PSA Signature Register. It is up to the CPU on how to deal with the PSA signature value just read. It can compare it to a known signature value stored at another memory location.

28.3.4 MSS_MCRC Registers

Table 28-50 lists the memory-mapped registers for the MSS_MCRC. All register offset addresses not listed in Table 28-50 should be considered as reserved locations and the register contents should not be modified.

Table 28-50. MSS_MCRC Registers

Offset	Acronym	Register Name	Section
0h	CRC_CTRL0	CRC Global Control Register 0	Section 28.3.4.1
8h	CRC_CTRL1	CRC Global Control Register 1	Section 28.3.4.2
10h	CRC_CTRL2	CRC Global Control Register 2	Section 28.3.4.3
18h	CRC_INTS	CRC Interrupt Enable Set Register	Section 28.3.4.4
20h	CRC_INTR	CRC Interrupt Enable Reset Register	Section 28.3.4.5
28h	CRC_STATUS_REG	CRC Interrupt Status Register-	Section 28.3.4.6
30h	CRC_INT_OFFSET_REG	CRC Interrupt Offset	Section 28.3.4.7
38h	CRC_BUSY	CRC Busy Register during AUTO mode	Section 28.3.4.8
40h	CRC_PCOUNT_REG1	CRC Pattern Counter Pre-load Register1	Section 28.3.4.9
44h	CRC_SCOUNT_REG1	CRC Sector Counter Pre-load Register1	Section 28.3.4.10
48h	CRC_CURSEC_REG1	CRC Current Sector Regis-ter 1	Section 28.3.4.11
4Ch	CRC_WDTOPLD1	CRC channel 1 Watchdog Timeout Preload Register A	Section 28.3.4.12
50h	CRC_BCTOPLD1	CRC channel 1 Block Complete Timeout Preload Register B	Section 28.3.4.13
60h	PSA_SIGREGL1	Channel 1 PSA signature low register	Section 28.3.4.14
64h	PSA_SIGREGH1	Channel 1 PSA signature high register	Section 28.3.4.15
68h	CRC_REGL1	Channel 1 CRC value low register	Section 28.3.4.16
6Ch	CRC_REGH1	Channel 1 CRC value high register	Section 28.3.4.17
70h	PSA_SECSIGREGL1	Channel 1 PSA sector sig-nature low register	Section 28.3.4.18
74h	PSA_SECSIGREGH1	Channel 1 PSA sector sig-nature high register	Section 28.3.4.19
78h	RAW_DATAREGL1	Channel 1 Raw Data Low Register	Section 28.3.4.20
7Ch	RAW_DATAREGH1	Channel 1 Raw Data High Register	Section 28.3.4.21
80h	CRC_PCOUNT_REG2	CRC Pattern Counter Pre-load Register2	Section 28.3.4.22
84h	CRC_SCOUNT_REG2	CRC Sector Counter Pre-load Register2	Section 28.3.4.23
88h	CRC_CURSEC_REG2	CRC Current Sector Regis-ter 2	Section 28.3.4.24
8Ch	CRC_WDTOPLD2	CRC channel 2 Watchdog Timeout Preload Register	Section 28.3.4.25
90h	CRC_BCTOPLD2	CRC channel 2 Block Com-plete Timeout Preload Reg-ister	Section 28.3.4.26
A0h	PSA_SIGREGL2	Channel 2 PSA signature low register	Section 28.3.4.27
A4h	PSA_SIGREGH2	Channel 2 PSA signature high register	Section 28.3.4.28
A8h	CRC_REGL2	Channel 2 CRC value low register	Section 28.3.4.29
ACh	CRC_REGH2	Channel 2 CRC value high register	Section 28.3.4.30
B0h	PSA_SECSIGREGL2	Channel 2 PSA sector sig-nature low register	Section 28.3.4.31
B4h	PSA_SECSIGREGH2	Channel 2 PSA sector sig-nature high register	Section 28.3.4.32
B8h	RAW_DATAREGL2	Channel 2 Raw Data Low Register	Section 28.3.4.33
BCh	RAW_DATAREGH2	Channel 2 Raw Data High register	Section 28.3.4.34
C0h	CRC_PCOUNT_REG3	CRC Pattern Counter Pre-load Register3	Section 28.3.4.35
C4h	CRC_SCOUNT_REG3	CRC Sector Counter Pre-load Register3	Section 28.3.4.36
C8h	CRC_CURSEC_REG3	CRC Current Sector Regis-ter 3	Section 28.3.4.37
CCh	CRC_WDTOPLD3	CRC channel 3 Watchdog Timeout Preload Register	Section 28.3.4.38
D0h	CRC_BCTOPLD3	CRC channel 3 Block Com-plete Timeout Preload Reg-ister	Section 28.3.4.39
E0h	PSA_SIGREGL3	Channel 3 PSA signature low register	Section 28.3.4.40
E4h	PSA_SIGREGH3	Channel 3 PSA signature high register	Section 28.3.4.41

Table 28-50. MSS_MCRC Registers (continued)

Offset	Acronym	Register Name	Section
E8h	CRC_REGL3	Channel 3 CRC value low register	Section 28.3.4.42
ECh	CRC_REGH3	Channel 3 CRC value high register	Section 28.3.4.43
F0h	PSA_SECSIGREGL3	Channel 3 PSA sector sig-nature low register	Section 28.3.4.44
F4h	PSA_SECSIGREGH3	Channel 3 PSA sector sig-nature high register	Section 28.3.4.45
F8h	RAW_DATAREGL3	Channel 3 Raw Data Low Register	Section 28.3.4.46
FCh	RAW_DATAREGH3	Channel 3 Raw Data High register	Section 28.3.4.47
100h	CRC_PCOUNT_REG4	CRC Pattern Counter Pre-load Register4	Section 28.3.4.48
104h	CRC_SCOUNT_REG4	CRC Sector Counter Pre-load Register4	Section 28.3.4.49
108h	CRC_CURSEC_REG4	CRC Current Sector Regis-ter 4	Section 28.3.4.50
10Ch	CRC_WDTOPLD4	CRC channel 4 Watchdog Timeout Preload Register	Section 28.3.4.51
110h	CRC_BCTOPLD4	CRC channel 4 Block Com-plete Timeout Preload Reg-ister	Section 28.3.4.52
120h	PSA_SIGREGL4	Channel 4 PSA signature low register	Section 28.3.4.53
124h	PSA_SIGREGH4	Channel 4 PSA signature high register	Section 28.3.4.54
128h	CRC_REGL4	Channel 4 CRC value low register	Section 28.3.4.55
12Ch	CRC_REGH4	Channel 4 CRC value high register	Section 28.3.4.56
130h	PSA_SECSIGREGL4	Channel 4 PSA sector sig-nature low register	Section 28.3.4.57
134h	PSA_SECSIGREGH4	Channel 4 PSA sector sig-nature high register	Section 28.3.4.58
138h	RAW_DATAREGL4	Channel 4 Raw Data Low Register	Section 28.3.4.59
13Ch	RAW_DATAREGH4	Channel 4 Raw Data High register	Section 28.3.4.60
140h	MCRC_BUS_SEL	Data bus tracing selection	Section 28.3.4.61

Complex bit access types are encoded to fit into small table cells. Table 28-51 shows the codes that are used for access types in this section.

Table 28-51. MSS_MCRC Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

28.3.4.1 CRC_CTRL0 Register (Offset = 0h) [reset = 0h]

CRC_CTRL0 is shown in Figure 28-66 and described in Table 28-52.

Return to Summary Table.

Contains sw reset control bit to reset PSA

Figure 28-66. CRC_CTRL0 Register

31	30	29	28	27	26	25	24
CH4_CRC_SEL2	CH4_BYTE_SWAP	CH4_BIT_SWAP	CH4_CRC_SEL		CH4_DW_SEL		CH4_PSA_SWREST
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h
23	22	21	20	19	18	17	16
CH3_CRC_SEL2	CH3_BYTE_SWAP	CH3_BIT_SWAP	CH3_CRC_SEL		CH3_DW_SEL		CH3_PSA_SWREST
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h
15	14	13	12	11	10	9	8
CH2_CRC_SEL2	CH2_BYTE_SWAP	CH2_BIT_SWAP	CH2_CRC_SEL		CH2_DW_SEL		CH2_PSA_SWREST
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
CH1_CRC_SEL2	CH1_BYTE_SWAP	CH1_BIT_SWAP	CH1_CRC_SEL		CH1_DW_SEL		CH1_PSA_SWREST
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h

Table 28-52. CRC_CTRL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CH4_CRC_SEL2	R/W	0h	Refer "CH4_DW_SEL" field description
30	CH4_BYTE_SWAP	R/W	0h	BYTE SWAP Enable across Data Size 0 – Byte Swap Disabled 1 – Byte Swap enabled.
29	CH4_BIT_SWAP	R/W	0h	msb/lsb SWAPPING 0 – msb (most significant bit First) 1 – lsb (least significant bit First)
28-27	CH4_CRC_SEL	R/W	0h	CRC type select. {CH1_CRC_SEL2,CH1_CRC_SEL[1:0]} 000 – CRC-64 001 - CRC-16 010 – CRC-32 100 - VDA CAN, SAE-J1850 CRC-8 101 - H2F, Autosar 4.0 110 - CASTAGNOLI, iSCSI 111 / 011 - E2E Profile 4
26-25	CH4_DW_SEL	R/W	0h	CRC Data Size select. 000 – 64 bit Data Size 001 - 16 bit Data Size 010 – 32 Bit Data Size
24	CH4_PSA_SWREST	R/W	0h	Channel 4 PSA Software Reset. When set, the PSA Signature Register is reset to all zero. Software reset does not reset software reset bit itself. Therefore, CPU is required to clear this bit by writing a '0'. 0 = PSA Signature Register not reset 1 = PSA Signature Register reset
23	CH3_CRC_SEL2	R/W	0h	Refer "CH3_DW_SEL" field description
22	CH3_BYTE_SWAP	R/W	0h	BYTE SWAP Enable across Data Size 0 – Byte Swap Disabled 1 – Byte Swap enabled.
21	CH3_BIT_SWAP	R/W	0h	msb/lsb SWAPPING 0 – msb (most significant bit First) 1 – lsb (least significant bit First)
20-19	CH3_CRC_SEL	R/W	0h	CRC type select. {CH1_CRC_SEL2,CH1_CRC_SEL[1:0]} 000 – CRC-64 001 - CRC-16 010 – CRC-32 100 - VDA CAN, SAE-J1850 CRC-8 101 - H2F, Autosar 4.0 110 - CASTAGNOLI, iSCSI 111 / 011 - E2E Profile 4
18-17	CH3_DW_SEL	R/W	0h	CRC Data Size select. 000 – 64 bit Data Size 001 - 16 bit Data Size 010 – 32 Bit Data Size

Table 28-52. CRC_CTRL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CH3_PSA_SWREST	R/W	0h	Channel 3 PSA Software Reset. When set, the PSA Signature Register is reset to all zero. Software reset does not reset software reset bit itself. Therefore, CPU is required to clear this bit by writing a '0'. 0 = PSA Signature Register not reset 1 = PSA Signature Register reset
15	CH2_CRC_SEL2	R/W	0h	Refer "CH2_DW_SEL" field description
14	CH2_BYTE_SWAP	R/W	0h	BYTE SWAP Enable across Data Size 0 – Byte Swap Disabled 1 – Byte Swap enabled.
13	CH2_BIT_SWAP	R/W	0h	msb/lsb SWAPPING 0 – msb (most significant bit First) 1 – lsb (least significant bit First)
12-11	CH2_CRC_SEL	R/W	0h	CRC type select. {CH1_CRC_SEL2,CH1_CRC_SEL[1:0]} 000 – CRC-64 001 - CRC-16 010 – CRC-32 100 - VDA CAN, SAE-J1850 CRC-8 101 - H2F, Autosar 4.0 110 - CASTAGNOLI, iSCSI 111 / 011 - E2E Profile 4
10-9	CH2_DW_SEL	R/W	0h	CRC Data Size select. 000 – 64 bit Data Size 001 - 16 bit Data Size 010 – 32 Bit Data Size
8	CH2_PSA_SWREST	R/W	0h	Channel 2 PSA Software Reset. When set, the PSA Signature Register is reset to all zero. Software reset does not reset software reset bit itself. Therefore, CPU is required to clear this bit by writing a '0'. 0 = PSA Signature Register not reset 1 = PSA Signature Register reset
7	CH1_CRC_SEL2	R/W	0h	Refer "CH1_DW_SEL" field description
6	CH1_BYTE_SWAP	R/W	0h	BYTE SWAP Enable across Data Size 0 – Byte Swap Disabled 1 – Byte Swap enabled.
5	CH1_BIT_SWAP	R/W	0h	msb/lsb SWAPPING 0 – msb (most significant bit First) 1 – lsb (least significant bit First)
4-3	CH1_CRC_SEL	R/W	0h	CRC type select. {CH1_CRC_SEL2,CH1_CRC_SEL[1:0]} 000 – CRC-64 001 - CRC-16 010 – CRC-32 100 - VDA CAN, SAE-J1850 CRC-8 101 - H2F, Autosar 4.0 110 - CASTAGNOLI, iSCSI 111 / 011 - E2E Profile 4
2-1	CH1_DW_SEL	R/W	0h	CRC Data Size select. 000 – 64 bit Data Size 001 - 16 bit Data Size 010 – 32 Bit Data Size
0	CH1_PSA_SWREST	R/W	0h	Channel 1 PSA Software Reset. When set, the PSA Signature Register is reset to all zero. Software reset does not reset software reset bit itself. Therefore, CPU is required to clear this bit by writing a '0'. 0 = PSA Signature Register not reset 1 = PSA Signature Register reset

28.3.4.2 CRC_CTRL1 Register (Offset = 8h) [reset = 0h]

CRC_CTRL1 is shown in Figure 28-67 and described in Table 28-53.

Return to Summary Table.

Contains power down control bit

Figure 28-67. CRC_CTRL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							PWDN
R-0h							R/W-0h

Table 28-53. CRC_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	PWDN	R/W	0h	Power Down. When set, MCRC moduleMCRC Module is put in power down mode. 0 = MCRC is not in power down mode 1 = MCRC is in power down mode

28.3.4.3 CRC_CTRL2 Register (Offset = 10h) [reset = 0h]

CRC_CTRL2 is shown in Figure 28-68 and described in Table 28-54.

Return to Summary Table.

Contains channel mode, data trace enable control bits

Figure 28-68. CRC_CTRL2 Register

31	30	29	28	27	26	25	24
RESERVED						CH4_MODE	
R-0h						R/W-0h	
23	22	21	20	19	18	17	16
RESERVED						CH3_MODE	
R-0h						R/W-0h	
15	14	13	12	11	10	9	8
RESERVED						CH2_MODE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED			CH1_TRACEEN	RESERVED		CH1_MODE	
R-0h			R/W-0h	R-0h		R/W-0h	

Table 28-54. CRC_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-24	CH4_MODE	R/W	0h	Channel 4 Mode: 0 0 = Data Capture mode. In this mode, the PSA Signature Register does not compress data when it is written. Any data written to PSA Signature Register is simply captured by PSA Signature Register without any compression. This mode can be used to plant seed value into the PSA register 0 1 = AUTO mode 1 0 = reserved 1 1 = Full-CPU mode
23-18	RESERVED	R	0h	
17-16	CH3_MODE	R/W	0h	Channel 3 Mode: 0 0 = Data Capture mode. In this mode, the PSA Signature Register does not compress data when it is written. Any data written to PSA Signature Register is simply captured by PSA Signature Register without any compression. This mode can be used to plant seed value into the PSA register 0 1 = AUTO mode 1 0 = reserved 1 1 = Full-CPU mode
15-10	RESERVED	R	0h	
9-8	CH2_MODE	R/W	0h	Channel 2 Mode: 0 0 = Data Capture mode. In this mode, the PSA Signature Register does not compress data when it is written. Any data written to PSA Signature Register is simply captured by PSA Signature Register without any compression. This mode can be used to plant seed value into the PSA register 0 1 = AUTO mode 1 0 = reserved 1 1 = Full-CPU mode
7-5	RESERVED	R	0h	
4	CH1_TRACEEN	R/W	0h	Channel 1 Data Trace Enable. When set, the channel is put into data trace mode. The channel snoops on the CPU VBUSM, ITCM, DTCM buses for any read transaction. Any read data on these buses is compressed by the PSA Signature Register. When suspend is on, the PSA Signature Register does not compress any read data on these buses. 0 = Data Trace disable 1 = Data Trace enable
3-2	RESERVED	R	0h	
1-0	CH1_MODE	R/W	0h	Channel 1 Mode: 0 0 = Data Capture mode. In this mode, the PSA Signature Register does not compress data when it is written. Any data written to PSA Signature Register is simply captured by PSA Signature Register without any compression. This mode can be used to plant seed value into the PSA register 0 1 = AUTO mode 1 0 = reserved 1 1 = Full-CPU mode

28.3.4.4 CRC_INTS Register (Offset = 18h) [reset = 0h]

CRC_INTS is shown in Figure 28-69 and described in Table 28-55.

Return to Summary Table.

Write one to a bit to enable a interrupt

Figure 28-69. CRC_INTS Register

31	30	29	28	27	26	25	24
RESERVED			CH4_TIMEOUTENS	CH4_UNDERENS	CH4_OVERENS	CH4_CRCFAILENS	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED			CH3_TIMEOUTENS	CH3_UNDERENS	CH3_OVERENS	CH3_CRCFAILENS	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			CH2_TIMEOUTENS	CH2_UNDERENS	CH2_OVERENS	CH2_CRCFAILENS	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED			CH1_TIMEOUTENS	CH1_UNDERENS	CH1_OVERENS	CH1_CRCFAILENS	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 28-55. CRC_INTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	CH4_TIMEOUTENS	R/W	0h	Channel 4 Timeout Interrupt Enable Bit. Writing a one to this bit enable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt enable
27	CH4_UNDERENS	R/W	0h	Channel 4 Underrun Interrupt Enable Bit. Writing a one to this bit enable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt enable
26	CH4_OVERENS	R/W	0h	Channel 4 Overrun Interrupt Enable Bit. Writing a one to this bit enable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt enable
25	CH4_CRCFAILENS	R/W	0h	Channel 4 CRC Fail Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt enable
24-21	RESERVED	R	0h	
20	CH3_TIMEOUTENS	R/W	0h	Channel 3 Timeout Interrupt Enable Bit. Writing a one to this bit enable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt enable

Table 28-55. CRC_INTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CH3_UNDERENS	R/W	0h	Channel 3 Underrun Interrupt Enable Bit. Writing a one to this bit enable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt enable
18	CH3_OVERENS	R/W	0h	Channel 3 Overrun Interrupt Enable Bit. Writing a one to this bit enable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt enable
17	CH3_CRCFAILENS	R/W	0h	Channel 3 CRC Fail Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt enable
16-13	RESERVED	R	0h	
12	CH2_TIMEOUTENS	R/W	0h	Channel 2 Timeout Interrupt Enable Bit. Writing a one to this bit enable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt enable
11	CH2_UNDERENS	R/W	0h	Channel 2 Underrun Interrupt Enable Bit. Writing a one to this bit enable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt enable
10	CH2_OVERENS	R/W	0h	Channel 2 Overrun Interrupt Enable Bit. Writing a one to this bit enable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt enable
9	CH2_CRCFAILENS	R/W	0h	Channel 2 CRC Fail Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt enable
8-5	RESERVED	R	0h	
4	CH1_TIMEOUTENS	R/W	0h	Channel 1 Timeout Interrupt Enable Bit. Writing a one to this bit enable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt enable
3	CH1_UNDERENS	R/W	0h	Channel 1 Underrun Interrupt Enable Bit. Writing a one to this bit enable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt enable

Table 28-55. CRC_INTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CH1_OVERENS	R/W	0h	Channel 1 Overrun Interrupt Enable Bit. Writing a one to this bit enable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt enable
1	CH1_CRCFAILENS	R/W	0h	Channel 1 CRC Fail Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt enable
0	RESERVED	R	0h	

28.3.4.5 CRC_INTR Register (Offset = 20h) [reset = 0h]

CRC_INTR is shown in Figure 28-70 and described in Table 28-56.

Return to Summary Table.

Write one to a bit to disable a interrupt

Figure 28-70. CRC_INTR Register

31	30	29	28	27	26	25	24
RESERVED			CH4_TIMEOUTENR	CH4_UNDERENR	CH4_OVERENR	CH4_CRCFAILENR	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED			CH3_TIMEOUTENR	CH3_UNDERENR	CH3_OVERENR	CH3_CRCFAILENR	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			CH2_TIMEOUTENR	CH2_UNDERENR	CH2_OVERENR	CH2_CRCFAILENR	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED			CH1_TIMEOUTENR	CH1_UNDERENR	CH1_OVERENR	CH1_CRCFAILENR	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 28-56. CRC_INTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	CH4_TIMEOUTENR	R/W	0h	Channel 4 Timeout Interrupt Disable Bit. Writing a one to this bit disable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt disable
27	CH4_UNDERENR	R/W	0h	Channel 4 Underrun Interrupt Disable Bit. Writing a one to this bit disable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/dis- able). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt disable
26	CH4_OVERENR	R/W	0h	Channel 4 Overrun Interrupt Disable Bit. Writing a one to this bit disable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt disable
25	CH4_CRCFAILENR	R/W	0h	Channel 4 CRC Fail Interrupt Disable Bit. Writing a one to this bit disable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt disable
24-21	RESERVED	R	0h	
20	CH3_TIMEOUTENR	R/W	0h	Channel 3 Timeout Interrupt Disable Bit. Writing a one to this bit disable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt disable

Table 28-56. CRC_INTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CH3_UNDERENR	R/W	0h	Channel 3 Underrun Interrupt Disable Bit. Writing a one to this bit disable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/dis- able). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt disable
18	CH3_OVERENR	R/W	0h	Channel 3 Overrun Interrupt Disable Bit. Writing a one to this bit disable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt disable
17	CH3_CRCFAILENR	R/W	0h	Channel 3 CRC Fail Interrupt Disable Bit. Writing a one to this bit disable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt disable
16-13	RESERVED	R	0h	
12	CH2_TIMEOUTENR	R/W	0h	Channel 2 Timeout Interrupt Disable Bit. Writing a one to this bit disable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt disable
11	CH2_UNDERENR	R/W	0h	Channel 2 Underrun Interrupt Disable Bit. Writing a one to this bit disable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/dis- able). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt disable
10	CH2_OVERENR	R/W	0h	Channel 2 Overrun Interrupt Disable Bit. Writing a one to this bit disable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt disable
9	CH2_CRCFAILENR	R/W	0h	Channel 2 CRC Fail Interrupt Disable Bit. Writing a one to this bit disable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt disable
8-5	RESERVED	R	0h	
4	CH1_TIMEOUTENR	R/W	0h	Channel 1 Timeout Interrupt Disable Bit. Writing a one to this bit disable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt disable
3	CH1_UNDERENR	R/W	0h	Channel 1 Underrun Interrupt Disable Bit. Writing a one to this bit disable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/dis- able). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt disable

Table 28-56. CRC_INTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CH1_OVERENR	R/W	0h	Channel 1 Overrun Interrupt Disable Bit. Writing a one to this bit disable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt disable
1	CH1_CRCFAILENR	R/W	0h	Channel 1 CRC Fail Interrupt Disable Bit. Writing a one to this bit disable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt disable
0	RESERVED	R	0h	

28.3.4.6 CRC_STATUS_REG Register (Offset = 28h) [reset = 0h]

CRC_STATUS_REG is shown in Figure 28-71 and described in Table 28-57.

Return to Summary Table.

Contains interrupt flags for different types of interrupt

Figure 28-71. CRC_STATUS_REG Register

31	30	29	28	27	26	25	24
RESERVED			CH4_TIMEOUT	CH4_UNDER	CH4_OVER	CH4_CRCFAIL	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED			CH3_TIMEOUT	CH3_UNDER	CH3_OVER	CH3_CRCFAIL	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			CH2_TIMEOUT	CH2_UNDER	CH2_OVER	CH2_CRCFAIL	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED			CH1_TIMEOUT	CH1_UNDER	CH1_OVER	CH1_CRCFAIL	RESERVED
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 28-57. CRC_STATUS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	CH4_TIMEOUT	R/W	0h	Channel 4 CRC Timeout Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode. 0 = No timeout interrupt is active 1 = Timeout interrupt is active
27	CH4_UNDER	R/W	0h	Channel 4 CRC Underrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only 0 = No underrun interrupt is active 1 = Underrun interrupt is active
26	CH4_OVER	R/W	0h	Channel 4 CRC Overrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode 0 = No overrun interrupt is active 1 = Overrun interrupt is active
25	CH4_CRCFAIL	R/W	0h	Channel 4 CRC Compare Fail Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only. 0 = No CRC compare fail interrupt is active 1 = CRC compare fail interrupt is active
24-21	RESERVED	R	0h	
20	CH3_TIMEOUT	R/W	0h	Channel 3 CRC Timeout Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode. 0 = No timeout interrupt is active 1 = Timeout interrupt is active
19	CH3_UNDER	R/W	0h	Channel 3 CRC Underrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only 0 = No underrun interrupt is active 1 = Underrun interrupt is active
18	CH3_OVER	R/W	0h	Channel 3 CRC Overrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode 0 = No overrun interrupt is active 1 = Overrun interrupt is active
17	CH3_CRCFAIL	R/W	0h	Channel 3 CRC Compare Fail Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only. 0 = No CRC compare fail interrupt is active 1 = CRC compare fail interrupt is active
16-13	RESERVED	R	0h	
12	CH2_TIMEOUT	R/W	0h	Channel 2 CRC Timeout Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode. 0 = No timeout interrupt is active 1 = Timeout interrupt is active

Table 28-57. CRC_STATUS_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CH2_UNDER	R/W	0h	Channel 2 CRC Underrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only 0 = No underrun interrupt is active 1 = Underrun interrupt is active
10	CH2_OVER	R/W	0h	Channel 2 CRC Overrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode 0 = No overrun interrupt is active 1 = Overrun interrupt is active
9	CH2_CRCFAIL	R/W	0h	Channel 2 CRC Compare Fail Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only. 0 = No CRC compare fail interrupt is active 1 = CRC compare fail interrupt is active
8-5	RESERVED	R	0h	
4	CH1_TIMEOUT	R/W	0h	Channel 1 CRC Timeout Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode. 0 = No timeout interrupt is active 1 = Timeout interrupt is active
3	CH1_UNDER	R/W	0h	Channel 1 CRC Underrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only 0 = No underrun interrupt is active 1 = Underrun interrupt is active
2	CH1_OVER	R/W	0h	Channel 1 CRC Overrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode 0 = No overrun interrupt is active 1 = Overrun interrupt is active
1	CH1_CRCFAIL	R/W	0h	Channel 1 CRC Compare Fail Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only. 0 = No CRC compare fail interrupt is active 1 = CRC compare fail interrupt is active
0	RESERVED	R	0h	

28.3.4.7 CRC_INT_OFFSET_REG Register (Offset = 30h) [reset = 0h]

CRC_INT_OFFSET_REG is shown in Figure 28-72 and described in Table 28-58.

Return to Summary Table.

Contains the interrupt offset vector address

Figure 28-72. CRC_INT_OFFSET_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OFSTREG							
R-0h																								R/W-0h							

Table 28-58. CRC_INT_OFFSET_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	OFSTREG	R/W	0h	CRC Interrupt Offset. This register indicates the highest priority pending interrupt vector address. Reading the offset register automatically clear the respective interrupt flag. Please reference Table 1–3. for details.

28.3.4.8 CRC_BUSY Register (Offset = 38h) [reset = 0h]

CRC_BUSY is shown in Figure 28-73 and described in Table 28-59.

Return to Summary Table.

Contains the busy flag for each channel

Figure 28-73. CRC_BUSY Register

31	30	29	28	27	26	25	24
RESERVED							Ch4_BUSY
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							Ch3_BUSY
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							Ch2_BUSY
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							CH1_BUSY
R-0h							R-0h

Table 28-59. CRC_BUSY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	Ch4_BUSY	R	0h	Ch4_BUSY. During AUTO mode, the busy flag is set when the first data pattern of the block is compressed and remains set until the the last data pattern of the block is compressed. The flag is cleared when the last data pattern of the block is compressed.
23-17	RESERVED	R	0h	
16	Ch3_BUSY	R	0h	Ch3_BUSY. During AUTO mode, the busy flag is set when the first data pattern of the block is compressed and remains set until the the last data pattern of the block is compressed. The flag is cleared when the last data pattern of the block is compressed.
15-9	RESERVED	R	0h	
8	Ch2_BUSY	R	0h	Ch2_BUSY. During AUTO mode, the busy flag is set when the first data pattern of the block is compressed and remains set until the the last data pattern of the block is compressed. The flag is cleared when the last data pattern of the block is compressed.
7-1	RESERVED	R	0h	
0	CH1_BUSY	R	0h	CH1_BUSY. During AUTO mode, the busy flag is set when the first data pattern of the block is compressed and remains set until the the last data pattern of the block is compressed. The flag is cleared when the last data pattern of the block is compressed.

28.3.4.9 CRC_PCOUNT_REG1 Register (Offset = 40h) [reset = 0h]

CRC_PCOUNT_REG1 is shown in Figure 28-74 and described in Table 28-60.

Return to Summary Table.

Channel 1 preload register for the pattern count

Figure 28-74. CRC_PCOUNT_REG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CRC_PAT_COUNT1																				
R-0h												R/W-0h																			

Table 28-60. CRC_PCOUNT_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	CRC_PAT_COUNT1	R/W	0h	Channel 1 Pattern Counter Preload Register. This register con- tains the number of data patterns in one sector to be compressed before a CRC is performed.

28.3.4.10 CRC_SCOUNT_REG1 Register (Offset = 44h) [reset = 0h]

CRC_SCOUNT_REG1 is shown in Figure 28-75 and described in Table 28-61.

Return to Summary Table.

Channel 1 preload register for the sector count

Figure 28-75. CRC_SCOUNT_REG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_SEC_COUNT1															
R-0h																R/W-0h															

Table 28-61. CRC_SCOUNT_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_SEC_COUNT1	R/W	0h	Channel 1 Sector Counter Preload Register. This register con- tains the number of sectors in one block of memory.

28.3.4.11 CRC_CURSEC_REG1 Register (Offset = 48h) [reset = 0h]

CRC_CURSEC_REG1 is shown in Figure 28-76 and described in Table 28-62.

Return to Summary Table.

Channel 1 current sector register contains the sector number which causes CRC failure

Figure 28-76. CRC_CURSEC_REG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_CURSEC1															
R-0h																R/W-0h															

Table 28-62. CRC_CURSEC_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_CURSEC1	R/W	0h	Channel 1 Current Sector ID Register. In AUTO mode, this register contains the current sector number of which the signature verification fails. The sector counter is a free running up counter. When a sector fails, the erroneous sector number is logged into current sector ID register and the CRC fail interrupt is generated The sector ID register is frozen until it is read and the CRC fail status bit is cleared by CPU. While it is frozen, it does not capture another erroneous sector number. When this condition happens, an overrun interrupt is generated instead. Once the register is read and the CRC fail interrupt flag is cleared it can capture new erro- neous sector number.

28.3.4.12 CRC_WDTOPLD1 Register (Offset = 4Ch) [reset = 0h]

CRC_WDTOPLD1 is shown in Figure 28-77 and described in Table 28-63.

Return to Summary Table.

Channel 1 timeout pre-load value to check if within a given time DMA initiates a block transfer

Figure 28-77. CRC_WDTOPLD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_WDTOPLD1																							
R-0h								R/W-0h																							

Table 28-63. CRC_WDTOPLD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_WDTOPLD1	R/W	0h	Channel 1 Watchdog Timeout Counter Preload Register. This register contains the number of clock cycles within which the DMA must transfer the next block of data patterns.

28.3.4.13 CRC_BCTOPLD1 Register (Offset = 50h) [reset = 0h]

CRC_BCTOPLD1 is shown in Figure 28-78 and described in Table 28-64.

Return to Summary Table.

Channel 1 timeout pre-load value to check if one block of patterns are compressed with a given time

Figure 28-78. CRC_BCTOPLD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_BCTOPLD1																							
R-0h								R/W-0h																							

Table 28-64. CRC_BCTOPLD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_BCTOPLD1	R/W	0h	Channel 1 Block Complete Timeout Counter Preload Regis- ter. This register contains the number of clock cycles within which the CRC for an entire block needs to complete before a timeout interrupt is generated.

28.3.4.14 PSA_SIGREGL1 Register (Offset = 60h) [reset = 0h]

PSA_SIGREGL1 is shown in Figure 28-79 and described in Table 28-65.

Return to Summary Table.

Channel 1 PSA signature low register

Figure 28-79. PSA_SIGREGL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASIG1_31_0																															
R/W-0h																															

Table 28-65. PSA_SIGREGL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASIG1_31_0	R/W	0h	Channel 1 PSA Signature Low Register. This register contains the value stored at PSASIG1[31:0] register.

28.3.4.15 PSA_SIGREGH1 Register (Offset = 64h) [reset = 0h]

PSA_SIGREGH1 is shown in Figure 28-80 and described in Table 28-66.

Return to Summary Table.

Channel 1 PSA signature high register

Figure 28-80. PSA_SIGREGH1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSA_SIG1_63_32																															
R/W-0h																															

Table 28-66. PSA_SIGREGH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSA_SIG1_63_32	R/W	0h	Channel 1 PSA Signature High Register. This register contains the value stored at PSASIG1[63:32] register.

28.3.4.16 CRC_REGL1 Register (Offset = 68h) [reset = 0h]

CRC_REGL1 is shown in Figure 28-81 and described in Table 28-67.

Return to Summary Table.

Channel 1 CRC value low register

Figure 28-81. CRC_REGL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC1_31_0																															
R/W-0h																															

Table 28-67. CRC_REGL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC1_31_0	R/W	0h	Channel 1 CRC Value Low Register. This register contains the current known good signature value stored at CRC1[31:0] regis- ter.

28.3.4.17 CRC_REGH1 Register (Offset = 6Ch) [reset = 0h]

CRC_REGH1 is shown in Figure 28-82 and described in Table 28-68.

Return to Summary Table.

Channel 1 CRC value high register

Figure 28-82. CRC_REGH1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC1_63_32																															
R/W-0h																															

Table 28-68. CRC_REGH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC1_63_32	R/W	0h	Channel 1 CRC Value High Register. This register contains the current known good signature value stored at CRC1[63:32] register.

28.3.4.18 PSA_SECSIGREGL1 Register (Offset = 70h) [reset = 0h]

PSA_SECSIGREGL1 is shown in Figure 28-83 and described in Table 28-69.

Return to Summary Table.

Channel 1 PSA sector signature low regis-ter

Figure 28-83. PSA_SECSIGREGL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG1_31_0																															
R-0h																															

Table 28-69. PSA_SECSIGREGL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG1_31_0	R	0h	Channel 1 PSA Sector Signature Low Register. This register contains the value stored at PSASECSIG1[31:0] register.

28.3.4.19 PSA_SECSIGREGH1 Register (Offset = 74h) [reset = 0h]

PSA_SECSIGREGH1 is shown in Figure 28-84 and described in Table 28-70.

Return to Summary Table.

Channel 1 PSA sector signature high regis-ter

Figure 28-84. PSA_SECSIGREGH1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG1_63_32																															
R-0h																															

Table 28-70. PSA_SECSIGREGH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG1_63_32	R	0h	Channel 1 PSA Sector Signature High Register. This register contains the value stored at PSASECSIG1[63:32] register.

28.3.4.20 RAW_DATAREGL1 Register (Offset = 78h) [reset = 0h]

RAW_DATAREGL1 is shown in Figure 28-85 and described in Table 28-71.

Return to Summary Table.

Channel 1 un-compressed raw data low register

Figure 28-85. RAW_DATAREGL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA1_31_0																															
R-0h																															

Table 28-71. RAW_DATAREGL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA1_31_0	R	0h	Channel 1 Raw Data Low Register. This register contains bit 31:0 of the un-compressed raw data.

28.3.4.21 RAW_DATAREGH1 Register (Offset = 7Ch) [reset = 0h]

RAW_DATAREGH1 is shown in Figure 28-86 and described in Table 28-72.

Return to Summary Table.

Channel 1 un-compressed raw data high register

Figure 28-86. RAW_DATAREGH1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA1_63_32																															
R-0h																															

Table 28-72. RAW_DATAREGH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA1_63_32	R	0h	Channel 1 Raw Data High Register. This register contains bit 63:32 of the un-compressed raw data.

28.3.4.22 CRC_PCOUNT_REG2 Register (Offset = 80h) [reset = 0h]

CRC_PCOUNT_REG2 is shown in Figure 28-87 and described in Table 28-73.

Return to Summary Table.

Channel 2 preload register for the pattern count

Figure 28-87. CRC_PCOUNT_REG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CRC_PAT_COUNT2																				
R-0h												R/W-0h																			

Table 28-73. CRC_PCOUNT_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	CRC_PAT_COUNT2	R/W	0h	Channel 2 Pattern Counter Preload Register. This register con- tains the number of data patterns in one sector to be compressed before a CRC is performed.

28.3.4.23 CRC_SCOUNT_REG2 Register (Offset = 84h) [reset = 0h]

CRC_SCOUNT_REG2 is shown in Figure 28-88 and described in Table 28-74.

Return to Summary Table.

Channel 2 preload register for the sector count

Figure 28-88. CRC_SCOUNT_REG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_SEC_COUNT2															
R-0h																R/W-0h															

Table 28-74. CRC_SCOUNT_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_SEC_COUNT2	R/W	0h	Channel 2 Sector Counter Preload Register. This register con- tains the number of sectors in one block of memory.

28.3.4.24 CRC_CURSEC_REG2 Register (Offset = 88h) [reset = 0h]

CRC_CURSEC_REG2 is shown in Figure 28-89 and described in Table 28-75.

Return to Summary Table.

Channel 2 current sector register contains the sector number which causes CRC fail-ure

Figure 28-89. CRC_CURSEC_REG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_CURSEC2															
R-0h																R/W-0h															

Table 28-75. CRC_CURSEC_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_CURSEC2	R/W	0h	Channel 2 Current Sector ID Register. In AUTO mode, this register contains the current sector number of which the signature verification fails. The sector counter is a free running up counter. When a sector fails, the erroneous sector number is logged into current sector ID register and the CRC fail interrupt is generated The sector ID register is frozen until it is read and the CRC fail status bit is cleared by CPU. While it is frozen, it does not capture another erroneous sector number. When this condition happens, an overrun interrupt is generated instead. Once the register is read and the CRC fail interrupt flag is cleared it can capture new erro- neous sector number.

28.3.4.25 CRC_WDTOPLD2 Register (Offset = 8Ch) [reset = 0h]

CRC_WDTOPLD2 is shown in Figure 28-90 and described in Table 28-76.

Return to Summary Table.

Channel 2 timeout pre-load value to check if within a given time DMA initiates a block transfer

Figure 28-90. CRC_WDTOPLD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_WDTOPLD2																							
R-0h								R/W-0h																							

Table 28-76. CRC_WDTOPLD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_WDTOPLD2	R/W	0h	Channel 2 Watchdog Timeout Counter Preload Register. This register contains the number of clock cycles within which the DMA must transfer the next block of data patterns.

28.3.4.26 CRC_BCTOPLD2 Register (Offset = 90h) [reset = 0h]

CRC_BCTOPLD2 is shown in Figure 28-91 and described in Table 28-77.

Return to Summary Table.

Channel 2 timeout pre-load value to check if one block of patterns are compressed with a given time

Figure 28-91. CRC_BCTOPLD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_BCTOPLD2																							
R-0h								R/W-0h																							

Table 28-77. CRC_BCTOPLD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_BCTOPLD2	R/W	0h	Channel 2 Block Complete Timeout Counter Preload Regis- ter. This register contains the number of clock cycles within which the CRC for an entire block needs to complete before a timeout interrupt is generated.

28.3.4.27 PSA_SIGREGL2 Register (Offset = A0h) [reset = 0h]

PSA_SIGREGL2 is shown in Figure 28-92 and described in Table 28-78.

Return to Summary Table.

Channel 2 PSA signature low register

Figure 28-92. PSA_SIGREGL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASIG2_31_0																															
R/W-0h																															

Table 28-78. PSA_SIGREGL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASIG2_31_0	R/W	0h	Channel 2 PSA Signature Low Register. This register contains the value stored at PSASIG2[31:0] register.

28.3.4.28 PSA_SIGREGH2 Register (Offset = A4h) [reset = 0h]

PSA_SIGREGH2 is shown in Figure 28-93 and described in Table 28-79.

Return to Summary Table.

Channel 2 PSA signature high register

Figure 28-93. PSA_SIGREGH2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSA_SIG2_63_32																															
R/W-0h																															

Table 28-79. PSA_SIGREGH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSA_SIG2_63_32	R/W	0h	Channel 2 PSA Signature High Register. This register contains the value stored at PSASIG2[63:32] register.

28.3.4.29 CRC_REGL2 Register (Offset = A8h) [reset = 0h]

CRC_REGL2 is shown in Figure 28-94 and described in Table 28-80.

Return to Summary Table.

Channel 2 CRC value low register

Figure 28-94. CRC_REGL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC2_31_0																															
R/W-0h																															

Table 28-80. CRC_REGL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC2_31_0	R/W	0h	Channel 2 CRC Value Low Register. This register contains the current known good signature value stored at CRC2[31:0] regis- ter.

28.3.4.30 CRC_REGH2 Register (Offset = ACh) [reset = 0h]

CRC_REGH2 is shown in Figure 28-95 and described in Table 28-81.

Return to Summary Table.

Channel 2 CRC value high register

Figure 28-95. CRC_REGH2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC2_63_32																															
R/W-0h																															

Table 28-81. CRC_REGH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC2_63_32	R/W	0h	Channel 2 CRC Value High Register. This register contains the current known good signature value stored at CRC2[63:32] register.

28.3.4.31 PSA_SECSIGREGL2 Register (Offset = B0h) [reset = 0h]

PSA_SECSIGREGL2 is shown in Figure 28-96 and described in Table 28-82.

Return to Summary Table.

Channel 2 PSA sector signature low regis-ter

Figure 28-96. PSA_SECSIGREGL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG2_31_0																															
R-0h																															

Table 28-82. PSA_SECSIGREGL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG2_31_0	R	0h	Channel 2 PSA Sector Signature Low Register. This register contains the value stored at PSASECSIG2[31:0] register.

28.3.4.32 PSA_SECSIGREGH2 Register (Offset = B4h) [reset = 0h]

PSA_SECSIGREGH2 is shown in Figure 28-97 and described in Table 28-83.

Return to Summary Table.

Channel 2 PSA sector signature high regis-ter

Figure 28-97. PSA_SECSIGREGH2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG2_63_32																															
R-0h																															

Table 28-83. PSA_SECSIGREGH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG2_63_32	R	0h	Channel 2 PSA Sector Signature High Register. This register contains the value stored at PSASECSIG2[63:32] register.

28.3.4.33 RAW_DATAREGL2 Register (Offset = B8h) [reset = 0h]

RAW_DATAREGL2 is shown in Figure 28-98 and described in Table 28-84.

Return to Summary Table.

Channel 2 un-compressed raw data low register

Figure 28-98. RAW_DATAREGL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA2_31_0																															
R-0h																															

Table 28-84. RAW_DATAREGL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA2_31_0	R	0h	Channel 2 Raw Data Low Register. This register contains bit 31:0 of the un-compressed raw data.

28.3.4.34 RAW_DATAREGH2 Register (Offset = BCh) [reset = 0h]

RAW_DATAREGH2 is shown in Figure 28-99 and described in Table 28-85.

Return to Summary Table.

Channel 2 un-compressed raw data high Register

Figure 28-99. RAW_DATAREGH2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA2_63_32																															
R-0h																															

Table 28-85. RAW_DATAREGH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA2_63_32	R	0h	Channel 2 Raw Data High Register. This register contains bit 63:32 of the un-compressed raw data.

28.3.4.35 CRC_PCOUNT_REG3 Register (Offset = C0h) [reset = 0h]

CRC_PCOUNT_REG3 is shown in Figure 28-100 and described in Table 28-86.

Return to Summary Table.

Channel 3 preload register for the pattern count

Figure 28-100. CRC_PCOUNT_REG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CRC_PAT_COUNT3																				
R-0h												R/W-0h																			

Table 28-86. CRC_PCOUNT_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	CRC_PAT_COUNT3	R/W	0h	Channel 3 Pattern Counter Preload Register. This register con- tains the number of data patterns in one sector to be compressed before a CRC is performed.

28.3.4.36 CRC_SCOUNT_REG3 Register (Offset = C4h) [reset = 0h]

CRC_SCOUNT_REG3 is shown in Figure 28-101 and described in Table 28-87.

Return to Summary Table.

Channel 3 preload register for the sector count

Figure 28-101. CRC_SCOUNT_REG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_SEC_COUNT3															
R-0h																R/W-0h															

Table 28-87. CRC_SCOUNT_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_SEC_COUNT3	R/W	0h	Channel 3 Sector Counter Preload Register. This register con- tains the number of sectors in one block of memory.

28.3.4.37 CRC_CURSEC_REG3 Register (Offset = C8h) [reset = 0h]

CRC_CURSEC_REG3 is shown in Figure 28-102 and described in Table 28-88.

Return to Summary Table.

Channel 3 current sector register contains the sector number which causes CRC fail-ure

Figure 28-102. CRC_CURSEC_REG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_CURSEC3															
R-0h																R/W-0h															

Table 28-88. CRC_CURSEC_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_CURSEC3	R/W	0h	Channel 3 Current Sector ID Register. In AUTO mode, this register contains the current sector number of which the signature verification fails. The sector counter is a free running up counter. When a sector fails, the erroneous sector number is logged into current sector ID register and the CRC fail interrupt is generated The sector ID register is frozen until it is read and the CRC fail status bit is cleared by CPU. While it is frozen, it does not capture another erroneous sector number. When this condition happens, an overrun interrupt is generated instead. Once the register is read and the CRC fail interrupt flag is cleared it can capture new erro- neous sector number.

28.3.4.38 CRC_WDTOPLD3 Register (Offset = CCh) [reset = 0h]

CRC_WDTOPLD3 is shown in Figure 28-103 and described in Table 28-89.

Return to Summary Table.

Channel 3 timeout pre-load value to check if within a given time DMA initiates a block transfer

Figure 28-103. CRC_WDTOPLD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_WDTOPLD3																							
R-0h								R/W-0h																							

Table 28-89. CRC_WDTOPLD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_WDTOPLD3	R/W	0h	Channel 3 Watchdog Timeout Counter Preload Register. This register contains the number of clock cycles within which the DMA must transfer the next block of data patterns.

28.3.4.39 CRC_BCTOPLD3 Register (Offset = D0h) [reset = 0h]

CRC_BCTOPLD3 is shown in Figure 28-104 and described in Table 28-90.

Return to Summary Table.

Channel 3 timeout pre-load value to check if one block of patterns are compressed with a given time

Figure 28-104. CRC_BCTOPLD3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_BCTOPLD3																							
R-0h								R/W-0h																							

Table 28-90. CRC_BCTOPLD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_BCTOPLD3	R/W	0h	Channel 3 Block Complete Timeout Counter Preload Regis- ter. This register contains the number of clock cycles within which the CRC for an entire block needs to complete before a timeout interrupt is generated.

28.3.4.40 PSA_SIGREGL3 Register (Offset = E0h) [reset = 0h]

PSA_SIGREGL3 is shown in Figure 28-105 and described in Table 28-91.

Return to Summary Table.

Channel 3 PSA signature low register

Figure 28-105. PSA_SIGREGL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASIG3_31_0																															
R/W-0h																															

Table 28-91. PSA_SIGREGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASIG3_31_0	R/W	0h	Channel 3 PSA Signature Low Register. This register contains the value stored at PSASIG3[31:0] register.

28.3.4.41 PSA_SIGREGH3 Register (Offset = E4h) [reset = 0h]

PSA_SIGREGH3 is shown in Figure 28-106 and described in Table 28-92.

Return to Summary Table.

Channel 3 PSA signature high register

Figure 28-106. PSA_SIGREGH3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSA_SIG3_63_32																															
R/W-0h																															

Table 28-92. PSA_SIGREGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSA_SIG3_63_32	R/W	0h	Channel 3 PSA Signature High Register. This register contains the value stored at PSASIG3[63:32] register.

28.3.4.42 CRC_REGL3 Register (Offset = E8h) [reset = 0h]

CRC_REGL3 is shown in Figure 28-107 and described in Table 28-93.

Return to Summary Table.

Channel 3 CRC value low register

Figure 28-107. CRC_REGL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC3_31_0																															
R/W-0h																															

Table 28-93. CRC_REGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC3_31_0	R/W	0h	Channel 3 CRC Value Low Register. This register contains the current known good signature value stored at CRC3[31:0] regis- ter.

28.3.4.43 CRC_REGH3 Register (Offset = ECh) [reset = 0h]

CRC_REGH3 is shown in Figure 28-108 and described in Table 28-94.

Return to Summary Table.

Channel 3 CRC value high register

Figure 28-108. CRC_REGH3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC3_63_32																															
R/W-0h																															

Table 28-94. CRC_REGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC3_63_32	R/W	0h	Channel 3 CRC Value High Register. This register contains the current known good signature value stored at CRC3[63:32] register.

28.3.4.44 PSA_SECSIGREGL3 Register (Offset = F0h) [reset = 0h]

PSA_SECSIGREGL3 is shown in Figure 28-109 and described in Table 28-95.

Return to Summary Table.

Channel 3 PSA sector signature low regis-ter

Figure 28-109. PSA_SECSIGREGL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG3_31_0																															
R-0h																															

Table 28-95. PSA_SECSIGREGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG3_31_0	R	0h	Channel 3 PSA Sector Signature Low Register. This register contains the value stored at PSASECSIG3[31:0] register.

28.3.4.45 PSA_SECSIGREGH3 Register (Offset = F4h) [reset = 0h]

PSA_SECSIGREGH3 is shown in Figure 28-110 and described in Table 28-96.

Return to Summary Table.

Channel 3 PSA sector signature high regis-ter

Figure 28-110. PSA_SECSIGREGH3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG3_63_32																															
R-0h																															

Table 28-96. PSA_SECSIGREGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG3_63_32	R	0h	Channel 3 PSA Sector Signature High Register. This register contains the value stored at PSASECSIG3[63:32] register.

28.3.4.46 RAW_DATAREGL3 Register (Offset = F8h) [reset = 0h]

RAW_DATAREGL3 is shown in Figure 28-111 and described in Table 28-97.

Return to Summary Table.

Channel 3 un-compressed raw data low register

Figure 28-111. RAW_DATAREGL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA3_31_0																															
R-0h																															

Table 28-97. RAW_DATAREGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA3_31_0	R	0h	Channel 3 Raw Data Low Register. This register contains bit 31:0 of the un-compressed raw data.

28.3.4.47 RAW_DATAREGH3 Register (Offset = FCh) [reset = 0h]

RAW_DATAREGH3 is shown in Figure 28-112 and described in Table 28-98.

Return to Summary Table.

Channel 3 un-compressed raw data high Register

Figure 28-112. RAW_DATAREGH3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA3_63_32																															
R-0h																															

Table 28-98. RAW_DATAREGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA3_63_32	R	0h	Channel 3 Raw Data High Register. This register contains bit 63:32 of the un-compressed raw data.

28.3.4.48 CRC_PCOUNT_REG4 Register (Offset = 100h) [reset = 0h]

CRC_PCOUNT_REG4 is shown in Figure 28-113 and described in Table 28-99.

Return to Summary Table.

Channel 4 preload register for the pattern count

Figure 28-113. CRC_PCOUNT_REG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CRC_PAT_COUNT4																				
R-0h												R/W-0h																			

Table 28-99. CRC_PCOUNT_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	CRC_PAT_COUNT4	R/W	0h	Channel 4 Pattern Counter Preload Register. This register con- tains the number of data patterns in one sector to be compressed before a CRC is performed.

28.3.4.49 CRC_SCOUNT_REG4 Register (Offset = 104h) [reset = 0h]

CRC_SCOUNT_REG4 is shown in Figure 28-114 and described in Table 28-100.

Return to Summary Table.

Channel 4 preload register for the sector count

Figure 28-114. CRC_SCOUNT_REG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_SEC_COUNT4															
R-0h																R/W-0h															

Table 28-100. CRC_SCOUNT_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_SEC_COUNT4	R/W	0h	Channel 4 Sector Counter Preload Register. This register con- tains the number of sectors in one block of memory.

28.3.4.50 CRC_CURSEC_REG4 Register (Offset = 108h) [reset = 0h]

CRC_CURSEC_REG4 is shown in Figure 28-115 and described in Table 28-101.

Return to Summary Table.

Channel 4 current sector register contains the sector number which causes CRC fail-ure

Figure 28-115. CRC_CURSEC_REG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_CURSEC4															
R-0h																R/W-0h															

Table 28-101. CRC_CURSEC_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_CURSEC4	R/W	0h	Channel 4 Current Sector ID Register. In AUTO mode, this register contains the current sector number of which the signature verification fails. The sector counter is a free running up counter. When a sector fails, the erroneous sector number is logged into current sector ID register and the CRC fail interrupt is generated The sector ID register is frozen until it is read and the CRC fail status bit is cleared by CPU. While it is frozen, it does not capture another erroneous sector number. When this condition happens, an overrun interrupt is generated instead. Once the register is read and the CRC fail interrupt flag is cleared it can capture new erro- neous sector number.

28.3.4.51 CRC_WDTOPLD4 Register (Offset = 10Ch) [reset = 0h]

CRC_WDTOPLD4 is shown in Figure 28-116 and described in Table 28-102.

Return to Summary Table.

Channel 4 timeout pre-load value to check if within a given time DMA initiates a block transfer

Figure 28-116. CRC_WDTOPLD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_WDTOPLD4																							
R-0h								R/W-0h																							

Table 28-102. CRC_WDTOPLD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_WDTOPLD4	R/W	0h	Channel 4 Watchdog Timeout Counter Preload Register. This register contains the number of clock cycles within which the DMA must transfer the next block of data patterns.

28.3.4.52 CRC_BCTOPLD4 Register (Offset = 110h) [reset = 0h]

CRC_BCTOPLD4 is shown in Figure 28-117 and described in Table 28-103.

Return to Summary Table.

Channel 4 timeout pre-load value to check if one block of patterns are compressed with a given time

Figure 28-117. CRC_BCTOPLD4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_BCTOPLD4																							
R-0h								R/W-0h																							

Table 28-103. CRC_BCTOPLD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_BCTOPLD4	R/W	0h	Channel 4 Block Complete Timeout Counter Preload Regis- ter. This register contains the number of clock cycles within which the CRC for an entire block needs to complete before a timeout interrupt is generated.

28.3.4.53 PSA_SIGREGL4 Register (Offset = 120h) [reset = 0h]

PSA_SIGREGL4 is shown in Figure 28-118 and described in Table 28-104.

Return to Summary Table.

Channel 4 PSA signature low register

Figure 28-118. PSA_SIGREGL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASIG4_31_0																															
R/W-0h																															

Table 28-104. PSA_SIGREGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASIG4_31_0	R/W	0h	Channel 4 PSA Signature Low Register. This register contains the value stored at PSASIG4[31:0] register.

28.3.4.54 PSA_SIGREGH4 Register (Offset = 124h) [reset = 0h]

PSA_SIGREGH4 is shown in Figure 28-119 and described in Table 28-105.

Return to Summary Table.

Channel 4 PSA signature high register

Figure 28-119. PSA_SIGREGH4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSA_SIG4_63_32																															
R/W-0h																															

Table 28-105. PSA_SIGREGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSA_SIG4_63_32	R/W	0h	Channel 4 PSA Signature High Register. This register contains the value stored at PSASIG4[63:32] register.

28.3.4.55 CRC_REGL4 Register (Offset = 128h) [reset = 0h]

CRC_REGL4 is shown in Figure 28-120 and described in Table 28-106.

Return to Summary Table.

Channel 4 CRC value low register

Figure 28-120. CRC_REGL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC4_31_0																															
R/W-0h																															

Table 28-106. CRC_REGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC4_31_0	R/W	0h	Channel 4 CRC Value Low Register. This register contains the current known good signature value stored at CRC4[31:0] regis- ter.

28.3.4.56 CRC_REGH4 Register (Offset = 12Ch) [reset = 0h]

CRC_REGH4 is shown in Figure 28-121 and described in Table 28-107.

Return to Summary Table.

Channel 4 CRC value high register

Figure 28-121. CRC_REGH4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC4_63_32																															
R/W-0h																															

Table 28-107. CRC_REGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC4_63_32	R/W	0h	Channel 4 CRC Value High Register. This register contains the current known good signature value stored at CRC4[63:32] register.

28.3.4.57 PSA_SECSIGREGL4 Register (Offset = 130h) [reset = 0h]

PSA_SECSIGREGL4 is shown in Figure 28-122 and described in Table 28-108.

Return to Summary Table.

Channel 4 PSA sector signature low regis-ter

Figure 28-122. PSA_SECSIGREGL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG4_31_0																															
R-0h																															

Table 28-108. PSA_SECSIGREGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG4_31_0	R	0h	Channel 4 PSA Sector Signature Low Register. This register contains the value stored at PSASECSIG4[31:0] register.

28.3.4.58 PSA_SECSIGREGH4 Register (Offset = 134h) [reset = 0h]

PSA_SECSIGREGH4 is shown in Figure 28-123 and described in Table 28-109.

Return to Summary Table.

Channel 4 PSA sector signature high regis-ter

Figure 28-123. PSA_SECSIGREGH4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG4_63_32																															
R-0h																															

Table 28-109. PSA_SECSIGREGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PSASECSIG4_63_32	R	0h	Channel 4 PSA Sector Signature High Register. This register contains the value stored at PSASECSIG4[63:32] register.

28.3.4.59 RAW_DATAREGL4 Register (Offset = 138h) [reset = 0h]

RAW_DATAREGL4 is shown in Figure 28-124 and described in Table 28-110.

Return to Summary Table.

Channel 4 un-compressed raw data low register

Figure 28-124. RAW_DATAREGL4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA4_31_0																															
R-0h																															

Table 28-110. RAW_DATAREGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA4_31_0	R	0h	Channel 4 Raw Data Low Register. This register contains bit 31:0 of the un-compressed raw data.

28.3.4.60 RAW_DATAREGH4 Register (Offset = 13Ch) [reset = 0h]

RAW_DATAREGH4 is shown in Figure 28-125 and described in Table 28-111.

Return to Summary Table.

Channel 4 un-compressed raw data high Register

Figure 28-125. RAW_DATAREGH4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA4_63_32																															
R-0h																															

Table 28-111. RAW_DATAREGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_DATA4_63_32	R	0h	Channel 4 Raw Data High Register. This register contains bit 63:32 of the un-compressed raw data.

28.3.4.61 MCRC_BUS_SEL Register (Offset = 140h) [reset = 0h]

MCRC_BUS_SEL is shown in Figure 28-126 and described in Table 28-112.

Return to Summary Table.

Disables either or all tracing of data buses

Figure 28-126. MCRC_BUS_SEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					MEn	DTCMEn	ITCMEn
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 28-112. MCRC_BUS_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	MEn	R/W	0h	MEn. Enable/disables the tracing of VBUSM 0: Tracing of VBUSM master bus has been disabled 1: Tracing of VBUSM master bus has been enabled
1	DTCMEn	R/W	0h	DTCMEn. Enable/disables the tracing of data TCM 0: Tracing of DTCM_ODD and DTCM_EVEN buses have been disabled 1: Tracing of DTCM_ODD and DTCM_EVEN buses have been enabled
0	ITCMEn	R/W	0h	ITCMEn. Enable/disables the tracing of instruction TCM 0: Tracing of ITCM bus has been disabled 1: Tracing of ITCM bus has been enabled

28.4 Programmable Built-In Self-Test (PBIST)

This section describes the programmable built-in self-test (PBIST) controller module used for testing the on-chip memories.

28.4.1 Overview

The PBIST (Programmable Built-In Self-Test) controller architecture provides a run-time-programmable memory BIST engine for varying levels of coverage across many embedded memory instances.

Table 28-113. Register Base Address (16xx/18xx)

Name	Frame Address (Hex) Start	Frame Address (Hex) End	Size	Description
MSS_PBIST	0xFFFF_E400	0xFFFF_E5FF	64B	PBIST module configuration registers. This IP covers memory test for all memories excluding DSP L1P, L1D and associated TAG memories.
DSP_PBIST	0x01BD_0000	0x01BD_FFFF	64B	DSP PBIST module configuration registers. This IP covers memory test for DSP L1P, L1D and associated TAG memories.

NOTE: DSP_PBIST register space is accessible only by the DSP core.

Table 28-114. Register Base Address (14xx)

Name	Frame Address (Hex) Start	Frame Address (Hex) End	Size	Description
MSS_PBIST	0xFFFF_E400	0xFFFF_E5FF	464B	PBIST module configuration registers. This IP covers memory test for all memories excluding DSP L1P, L1D and associated TAG memories.

28.4.1.1 PBIST vs. Application Software-Based Testing

The PBIST architecture consists of a small coprocessor with a dedicated instruction set targeted specifically toward testing memories. This coprocessor executes test routines stored in the PBIST ROM and runs them on multiple on-chip memory instances. The on-chip memory configuration information is also stored in the PBIST ROM.

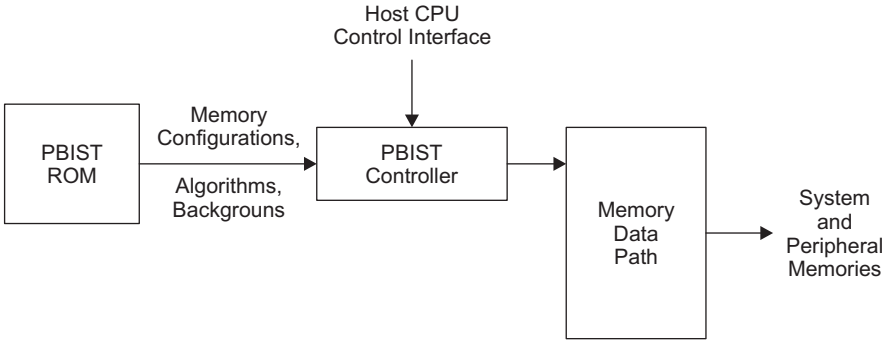
The PBIST Controller architecture offers significant advantages over tests running on the main Cortex-R4F processor (application software-based testing):

- Embedded CPUs have a long access path to memories outside the tightly-couple memory sub-system, while the PBIST controller has a dedicated path to the memories specifically for the self-test
- Embedded CPUs are designed for their targeted use and are often not easily programmed for memory test algorithms.
- The memory test algorithm code on embedded CPUs is typically significantly larger than that needed for PBIST.
- The embedded CPU is significantly larger than the PBIST controller.

28.4.1.2 PBIST Block Diagram

Figure 28-127 illustrates the basic PBIST blocks and its wrapper logic for the device.

Figure 28-127. PBIST Block Diagram



28.4.1.2.1 On-chip ROM

The on-chip ROM contains the information regarding the algorithms and memories to be tested.

28.4.1.2.2 Host Processor Interface to the PBIST Controller Registers

The Cortex-R4F CPU can select the algorithm and RAM groups for the memories' self-test from the on-chip ROM based on the application requirements. Once the self-test has executed, the CPU can query the PBIST controller registers to identify any memories that failed the self-test and to then take appropriate next steps as required by the application's author.

28.4.1.2.3 Memory Data Path

This is the read and write data path logic between different system and peripheral memories tightly coupled to the PBIST memory interface. The PBIST controller executes each selected algorithm on each valid memory group sequentially until all the algorithms are executed.

NOTE: Not all algorithms are designed to run on all RAM groups. If an algorithm is selected to run on an incompatible memory, this will result in a failure. Refer to and for RAM grouping and algorithm information.

28.4.2 RAM/ROM Grouping and Algorithm

28.4.2.1 RAM Algorithm: March13N

This section provides a brief description for some of the test algorithms used for memory self-test.

- **March13N:**
  - March13N is the baseline test algorithm for SRAM testing. It provides the highest overall coverage. The other algorithms provide additional coverage of otherwise missed boundary conditions of the SRAM operation.
  - The concept behind the general march algorithm is to indicate:
    - The bit cell can be written and read as both a 1 and a 0.
    - The bits around the bit cell do not affect the bit cell.
  - The basic operation of the march is to initialize the array to a know pattern, then march a different pattern through the memory.
  - Type of faults detected by this algorithm:
    - Address decoder faults
    - Stuck-At faults
    - Coupled faults
    - State coupling faults
    - Parametric faults
    - Write recovery faults
    - Read/write logic faults

28.4.2.2 ROM Algorithm: Triple Read XOR Read

The triple read reads the array, all the way through, three times while summing the reads to compare the sums for all three read formats. The algorithm checks if there is enough margin in both the erasure and programmation to operate at full speed with the CPU. This can be addressed with the XOR Read (Memory Contents XOR Memory Address). An error in the XOR Read indicates that the interaction between adjacent bit cells, being a different polarity, may be causing speed issues when the CPU exercises worst-case instruction sequencing. Each read can be performed on any memory block, and an associated checksum is calculated to determine PASS or FAIL.

Type of faults detected by this algorithm:

- Address decoder faults
- Stuck-At faults
- Coupled faults
- State coupling faults
- Parametric faults
- Read logic faults

NOTE: March13N is the most recommended algorithm for the memory self-test.

28.4.3 PBIST Registers

Table 28-115 lists the memory-mapped registers for the PBIST. All register offset addresses not listed in Table 28-115 should be considered as reserved locations and the register contents should not be modified.

Table 28-115. PBIST Registers

Offset	Acronym	Register Name	Section
164h	PBIST_DLR	Datalogger 0	Section 28.4.3.1
180h	PBIST_PACT	Pbist Active	Section 28.4.3.2
184h	PBIST_ID	PBIST ID	Section 28.4.3.3
188h	PBIST_OVR	PBIST Overrides	Section 28.4.3.4
190h	PBIST_FSFR0	Fail status fail - port 0	Section 28.4.3.5
194h	PBIST_FSFR1	Fail status fail - port 1	Section 28.4.3.6
198h	PBIST_FSRCR0	Fail status Count - port 0	Section 28.4.3.7
19Ch	PBIST_FSRCR1	Fail status Count - port 1	Section 28.4.3.8
1C0h	PBIST_ROM	Rom Mask	Section 28.4.3.9
1C4h	PBIST_ALGO	ROM Algorithm Mask 0	Section 28.4.3.10
1C8h	PBIST_RINFOL	RAM Info Mask Lower 0	Section 28.4.3.11
1CCh	PBIST_RINFOU	RAM Info Mask Upper 0	Section 28.4.3.12

Complex bit access types are encoded to fit into small table cells. Table 28-116 shows the codes that are used for access types in this section.

Table 28-116. PBIST Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

28.4.3.1 PBIST_DLR Register (Offset = 164h) [reset = 208h]

PBIST_DLR is shown in Figure 28-128 and described in Table 28-117.

Return to Summary Table.

Datalogger 0

Figure 28-128. PBIST_DLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DLR1								DLR0							
R/W-0h																R/W-2h								R/W-8h							

Table 28-117. PBIST_DLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved
15-8	DLR1	R/W	2h	Datalogger Register [8] : Reserevd [9] : Default Testing Mode. When in this mode, ROM-based testing is kicked off. If the intention is to perform go/no-go testing via config, write to both this bit and bit [2] of the Datalogger Register simultaneously [15:10] : Reserevd
7-0	DLR0	R/W	8h	Datalogger Register [1:0] : Reserved [2] : ROM-based testing mode. Setting this bit to 1 enables the PBIST controller to execute test algorithms that are stored in the PBIST ROM [3] : Do not change this bit from its default value of 1 [4] : Config access mode. Setting this bit allows the host processor to configure the PBIST controller registers [7:5] : Reserved

28.4.3.2 PBIST_PACT Register (Offset = 180h) [reset = 0h]

PBIST_PACT is shown in Figure 28-129 and described in Table 28-118.

Return to Summary Table.

Pbist Active

Figure 28-129. PBIST_PACT Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							PBIST_PACT
R/W-0h							R/W-0h

Table 28-118. PBIST_PACT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	Reserved
0	PBIST_PACT	R/W	0h	Pbist Active/ROM Clock Enable Register [0]: This bit must be set to turn on internal PBIST clocks. Setting this bit asserts an internal signal that is used as the clock gate enable. As long as this bit is 0, any access to PBIST will not go through, and PBIST will remain in an almost zero-power mode. Value 0 = Disable internal PBIST clocks Value 1 = Enable internal PBIST clocks

28.4.3.3 PBIST_ID Register (Offset = 184h) [reset = 1h]

PBIST_ID is shown in Figure 28-130 and described in Table 28-119.

Return to Summary Table.

PBIST ID

Figure 28-130. PBIST_ID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											PBIST_ID				
R/W-0h											R/W-1h				

Table 28-119. PBIST_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	Reserved
4-0	PBIST_ID	R/W	1h	PBIST ID. This is a unique ID assigned to each PBIST controller in a device with multiple PBIST controllers. The value of this register does not affect the functionality of the CPU interface.

28.4.3.4 PBIST_OVR Register (Offset = 188h) [reset = 9h]

PBIST_OVR is shown in Figure 28-131 and described in Table 28-120.

Return to Summary Table.

PBIST Overrides

Figure 28-131. PBIST_OVR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PBIST_OVR			
R/W-0h												R/W-9h			

Table 28-120. PBIST_OVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	Reserved
3-0	PBIST_OVR	R/W	9h	PBIST Overrides. [0] : RINFO Override Bit While doing ROM-based testing, each algorithm downloaded from the ROM has a memory mask associated with it that defines the applicable memory groups the algorithm will be run on. By default, this bit is set to 1, which means the memory mask that is downloaded from the ROM will overwrite the RAM info registers. The override bit can be reset by writing a 0 to it. In this case, the application can select the RAM groups to be tested by configuring the RAM info registers. NOTE: When this override bit = 0, each algorithm selected in ALGO register will run on each RAM selected in RINFOL and RINFOU register. It must be ensured that: a. Only the same type of memories (single port or two port) are selected, and b. Only memories that are valid for all algorithms enabled via the ALGO register are selected. If the above two requirements are not met, the memory self-test will fail. [3:1] : Reserved. This bit must not be changed from its default value of 0.

28.4.3.5 PBIST_FSFR0 Register (Offset = 190h) [reset = 0h]

PBIST_FSFR0 is shown in Figure 28-132 and described in Table 28-121.

Return to Summary Table.

Fail status fail - port 0

Figure 28-132. PBIST_FSFR0 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							PBIST_FSFR0
R-0h							R-0h

Table 28-121. PBIST_FSFR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PBIST_FSFR0	R	0h	Fail Status Fail Register- Port 0 This register indicates if a failure occurred during a memory self-test. Value 0 = No failure occurred Value 1 = Indicates a failure

28.4.3.6 PBIST_FSFR1 Register (Offset = 194h) [reset = 0h]

PBIST_FSFR1 is shown in Figure 28-133 and described in Table 28-122.

Return to Summary Table.

Fail status fail - port 1

Figure 28-133. PBIST_FSFR1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							PBIST_FSFR1
R-0h							R-0h

Table 28-122. PBIST_FSFR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PBIST_FSFR1	R	0h	Fail Status Fail Register- Port 1 This register indicates if a failure occurred during a memory self-test. Value 0 = No failure occurred Value 1 = Indicates a failure

28.4.3.7 PBIST_FSRCR0 Register (Offset = 198h) [reset = 0h]

PBIST_FSRCR0 is shown in Figure 28-134 and described in Table 28-123.

Return to Summary Table.

Fail Count fail - port 0

Figure 28-134. PBIST_FSRCR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PBIST_FSRCR0			
R-0h												R-0h			

Table 28-123. PBIST_FSRCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	PBIST_FSRCR0	R	0h	Fail Status Count - Port 0 These registers keep count of the number of failures observed during the memory self-test. The PBIST controller stops executing the memory self-test whenever a failure occurs in any memory instance for any of the test algorithms. The value in gets incremented by one whenever a failure occurs

28.4.3.8 PBIST_FSRCR1 Register (Offset = 19Ch) [reset = 0h]

PBIST_FSRCR1 is shown in Figure 28-135 and described in Table 28-124.

Return to Summary Table.

Fail Count fail - port 1

Figure 28-135. PBIST_FSRCR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PBIST_FSRCR1			
R-0h												R-0h			

Table 28-124. PBIST_FSRCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	PBIST_FSRCR1	R	0h	Fail Status Count - Port 1 These registers keep count of the number of failures observed during the memory self-test. The PBIST controller stops executing the memory self-test whenever a failure occurs in any memory instance for any of the test algorithms. The value in gets incremented by one whenever a failure occurs

28.4.3.9 PBIST_ROM Register (Offset = 1C0h) [reset = 3h]

PBIST_ROM is shown in Figure 28-136 and described in Table 28-125.

Return to Summary Table.

Rom Mask

Figure 28-136. PBIST_ROM Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						PBIST_ROM	
R/W-0h						R/W-3h	

Table 28-125. PBIST_ROM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	Reserved
1-0	PBIST_ROM	R/W	3h	Rom Mask . This two-bit register sets appropriate ROM access modes for the PBIST controller. Value 0h = No information is used from ROM Value 1h = Only RAM Group information from ROM Vaule 2h = Only Algorithm information from ROM Value 3h = Both Algorithm and RAM information from ROM. This option should be selected for application self-test.

28.4.3.10 PBIST_ALGO Register (Offset = 1C4h) [reset = FFFFFFFFh]

PBIST_ALGO is shown in Figure 28-137 and described in Table 28-126.

Return to Summary Table.

ROM Algorithm Mask 0

Figure 28-137. PBIST_ALGO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALGO3								ALGO2								ALGO1								ALGO0							
R/W-FFh								R/W-FFh								R/W-FFh								R/W-FFh							

Table 28-126. PBIST_ALGO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	ALGO3	R/W	FFh	This register is used to indicate the algorithms used for the memory self-test routine. Each bit corresponds to a specific algorithm. Writing a value 1 to the particular bit enables the corresponding algorithm. Writing a value 0 to the particular bit disables the corresponding algorithm.
23-16	ALGO2	R/W	FFh	This register is used to indicate the algorithms used for the memory self-test routine. Each bit corresponds to a specific algorithm. Writing a value 1 to the particular bit enables the corresponding algorithm. Writing a value 0 to the particular bit disables the corresponding algorithm.
15-8	ALGO1	R/W	FFh	This register is used to indicate the algorithms used for the memory self-test routine. Each bit corresponds to a specific algorithm. Writing a value 1 to the particular bit enables the corresponding algorithm. Writing a value 0 to the particular bit disables the corresponding algorithm.
7-0	ALGO0	R/W	FFh	This register is used to indicate the algorithms used for the memory self-test routine. Each bit corresponds to a specific algorithm. Writing a value 1 to the particular bit enables the corresponding algorithm. Writing a value 0 to the particular bit disables the corresponding algorithm.

28.4.3.11 PBIST_RINFOL Register (Offset = 1C8h) [reset = FFFFFFFFh]

PBIST_RINFOL is shown in Figure 28-138 and described in Table 28-127.

Return to Summary Table.

RAM Info Mask Lower 0

Figure 28-138. PBIST_RINFOL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RINFOL3								RINFOL2								RINFOL1								RINFOL0							
R/W-FFh								R/W-FFh								R/W-FFh								R/W-FFh							

Table 28-127. PBIST_RINFOL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RINFOL3	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.
23-16	RINFOL2	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.
15-8	RINFOL1	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.
7-0	RINFOL0	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.

28.4.3.12 PBIST_RINFOU Register (Offset = 1CCh) [reset = FFFFFFFFh]

PBIST_RINFOU is shown in Figure 28-139 and described in Table 28-128.

Return to Summary Table.

RAM Info Mask Upper 0

Figure 28-139. PBIST_RINFOU Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RINFOU3								RINFOU2								RINFOU1								RINFOU0							
R/W-FFh								R/W-FFh								R/W-FFh								R/W-FFh							

Table 28-128. PBIST_RINFOU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RINFOU3	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.
23-16	RINFOU2	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.
15-8	RINFOU1	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.
7-0	RINFOU0	R/W	FFh	This register is to select memory groups to run the algorithms selected in the PBIST_ALGO register. For an algorithm to be executed on a particular memory group, the corresponding bit in this register must be set to 1. The default value of this register is all 1s, which means all the memory groups are selected. Writing a value 0 to the particular bit disables the corresponding memory group.

28.5 Self-Test Controller (STC)

28.5.1 Integration Spec

28.5.1.1 Memory Map for 16xx/18xx

Table 28-129. Memory Map for 16xx/18xx

Name	Frame Address (Hex) Start	Frame Address (Hex) End	Size	Description
MSS_STC	0xFFFF_E600	0xFFFF_E7FF	284B	MSS_STC module configuration registers
DSS_STC	0x5004_0000	0x5004_011B	284B	DSS_STC module configuration registers

28.5.1.2 Memory Map for 14xx

Table 28-130. Memory Map for 14xx

Name	Frame Address (Hex) Start	Frame Address (Hex) End	Size	Description
MSS_STC	0xFFFF_E600	0xFFFF_E7FF	284B	MSS_STC module configuration registers

28.5.1.3 Features Not Supported in MSS_STC

- Section 28.5.6.1 – Launch-on-last-shift. TR_T =1
- Section 28.5.6.2 – Transition delay fault model. FT =1
- Section 28.5.6.7 – Low-power scan mode. MSS_STC.STCGCR1.LP_SCAN_MODE = 1
- Section 28.5.6.8 and Section 28.5.6.9 – Coverage improvement techniques – MSS_STC.STCGCR1.ROM_ACCESS_INV =1 mode
- Interval-based testing
- MSS_STC.STC_CLKDIV clock division features.

28.5.1.4 Features Not Supported in DSS_STC

- Section 28.5.6.1 – Launch-on-last-shift. TR_T =1
- Section 28.5.6.2 – Transition delay fault model. FT =1
- Section 28.5.6.6 – Non-low-power scan mode. DSS_STC.STCGCR1.LP_SCAN_MODE = 0
- Section 28.5.6.8 and Section 28.5.6.9 – Coverage improvement techniques – DSS_STC.STCGCR1.ROM_ACCESS_INV =1 mode
- Interval-based testing.
- DSS_STC.STC_CLKDIV clock division features.

28.5.2 General Description

The enhanced Self-Test Controller (STC) is used to test logic cores based on the On-Product Multiple Input Signature Register (OPMISR) scan compression architecture.

Software-based self-test programs for the cores are available, but offer less test coverage. Due to the complexity of the soft cores, it is difficult to achieve the required coverage; also, the program size is larger. For these complex cores, on-chip logic BIST support for the self-test is preferred.

The main features of this solution include:

- Implements the OPMISR controller, along with the on-chip self-test controller for the synthesizable module logic, which enables high test coverage.
- Can divide the complete test run into independent test sets (intervals).
- Capable of running the complete test, and running several intervals at a time.
- Can continue from the last executed interval (test set), and restart from the beginning (first interval in the ROM), or start from the first interval of each segment.
- A single self-test controller can support a test of up to 4 logical segments. A segment identifier corresponding to each interval is stored in the self-test ROM.
- The self-test controller facilitates complete isolation of the logical segment under the test from the rest of the system during the self-test run. Configure critical control signals in the master and slave ports of the logical segment under the test to a safe state.
- The self-tested CPU core master bus transaction signals are configured to be in idle mode during the self-test run.
- Can capture the failure interval number.
- Time-out counter for the self-test run as a fail-safe feature.
- Can read the MISR data (shifted from the OPMISR controller) of the last executed interval of the self-

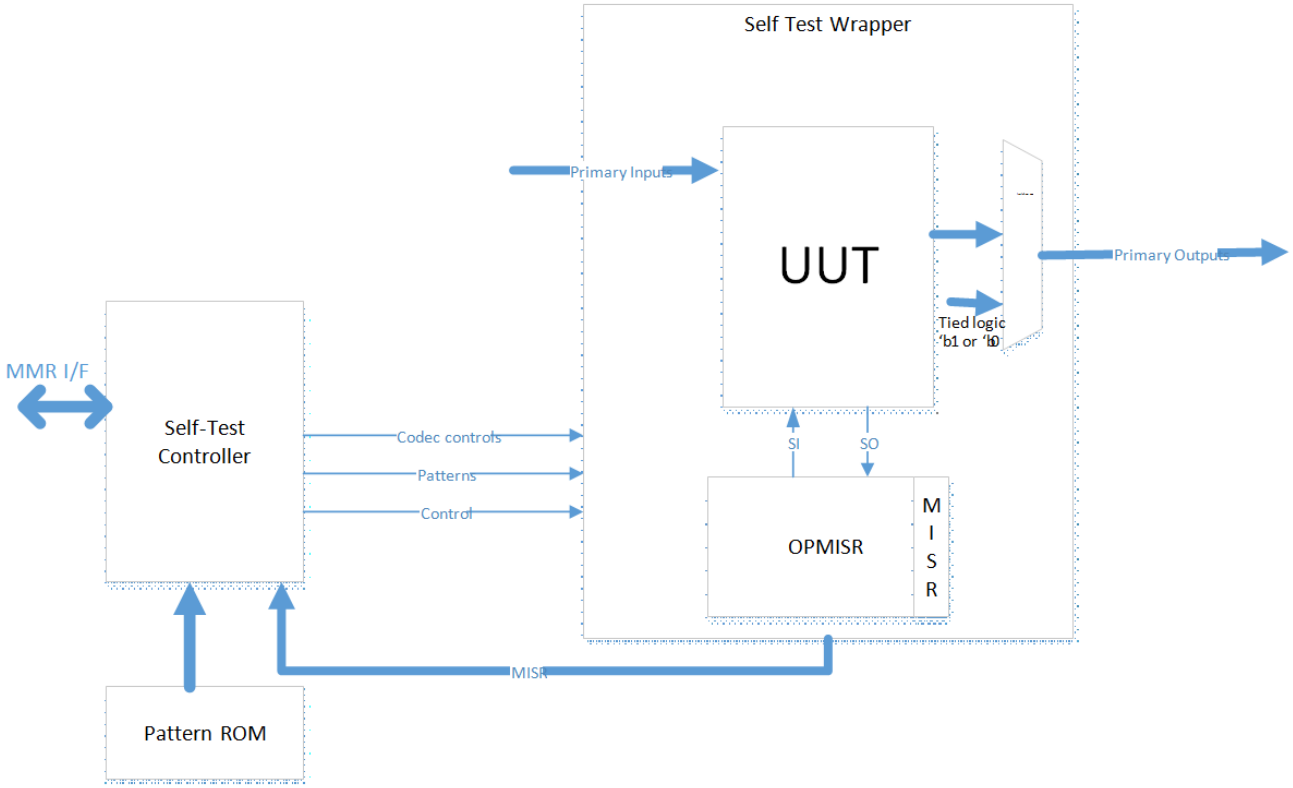
test run, for debugging purposes.

- Can capture power reduction using dead cycles before and after the capture pulse.
- Coverage improvements technique – ROM inverse access mode. In this, the patterns are read in a reverse order from ROM and applied to the UUT. Pattern randomization due to this approach results in coverage improvement, without an increase in the number of patterns. Corresponding INV_MISR is also stored in the ROM.

A self test segment corresponds to a portion of discreet safety-critical logic which can be tested in isolation from the rest of the system by the self test controller and OPMISR logic.

28.5.2.1 OPMISR Concept

Figure 28-140. OPMISR Conceptual Diagram



The On-Product Multiple-Input Signature Register (OPMISR) is a methodology which moves the test pattern generation on-chip. Logic BIST is implemented on functional partitions (BIST'ed COREs) that are speed-critical and have high gate count. A conceptual diagram of OPMISR implementation is shown in Figure 28-140.

The MISR test structure modifies the typical fullscan scan chain such that each scan data input internally drives many chains. These chains feed to the inserted MISR structure. The chain's values are captured into the MISR during shift, generating a resulting signature that can be shifted out.

A given Unit Under Test (UUT) is scan-inserted, and the scan chains are hooked to the OPMISR logic. The self-test wrapper created around the UUT and the OPMISR logic includes the isolation muxes for the output ports of the core, to ensure that the core and UUT are isolated from the rest of the system during the self-test.

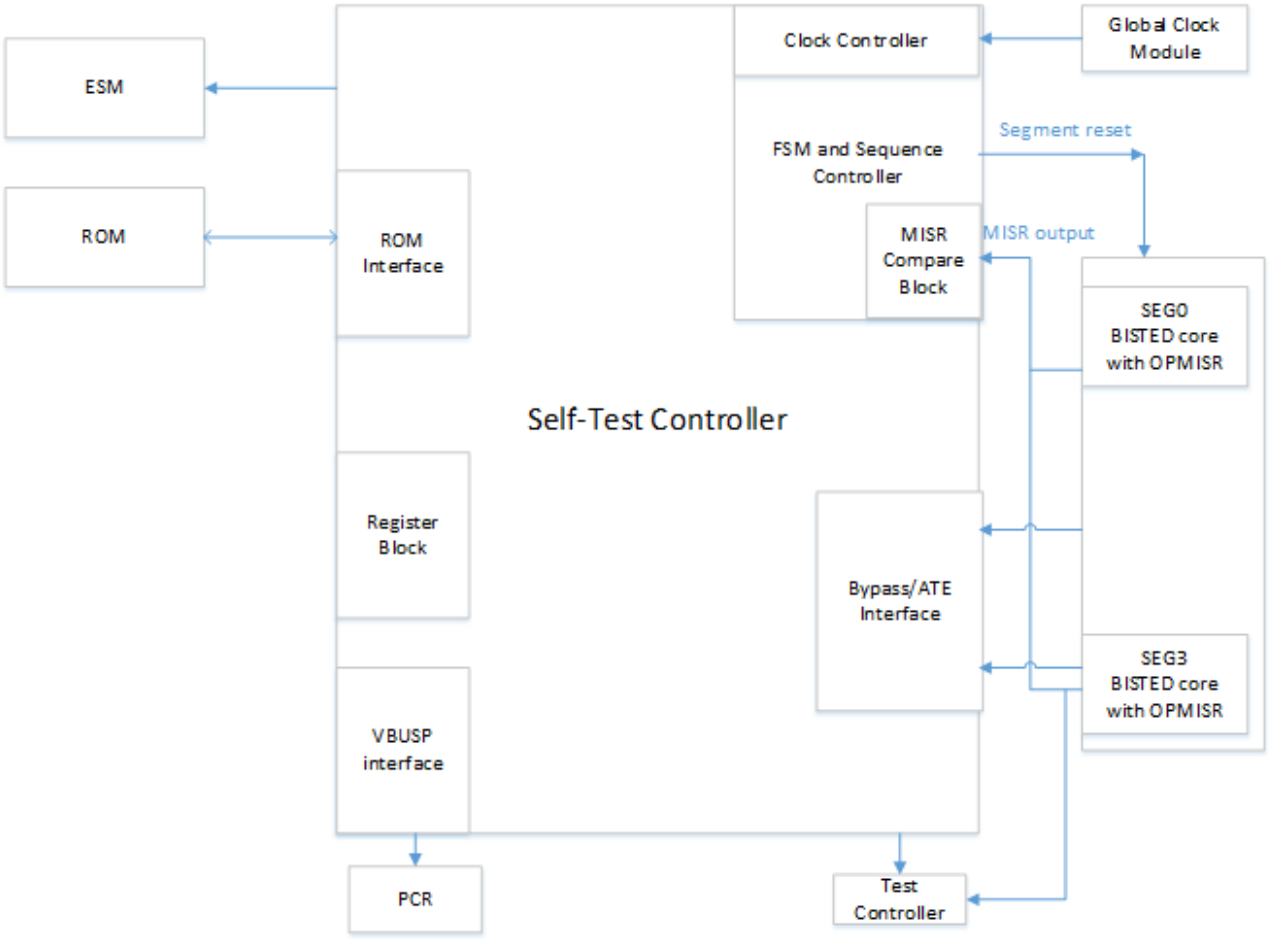
28.5.3 Block Diagram

The STC module is composed of following blocks:

- ROM interface

- FSM and sequence control
- Register file
- STC bypass / ATE interface
- Peripheral bus interface (VBUSP interface)

Figure 28-141. Block Diagram for STC With Multiple Segments



28.5.4 Module Description

28.5.4.1 ROM Interface

This block handles the ROM address and control signal generation to read the self-test microcode from the ROM. The test microcode, patterns, and golden signature value for each interval is stored in ROM. Detailed information of the ROM microcode is available at ROM.

28.5.4.2 FSM and Sequence Control

This block generates the signals and data to OPMISR controller based on the test type and scan chain depth. The sequence of operation per interval is defined in Section 28.5.5.

28.5.4.2.1 Clock Control

The CLOCK CNTRL sub-block handles the clock selection and clock generation for ROM, OPMISR controller, and BIST'ed CORE clocks.

28.5.4.2.2 MISR Compare Block

At the end of the each self-test interval, an 896-bit MISR value from the OPMISR controller is shifted into NSTC. This is compared with the MISR_GOLDEN value, which is copied into a buffered register before the start of the interval. The result is updated into the status registers.

28.5.4.3 Register Block

This block implements the user-programmable control registers that determine when to start a self test, at what clock frequency the scan test should be performed, which segment to be selected for the test, how many pattern intervals to be completed before stopping, and so forth.

The register block also captures various status information of the self test for the user.

28.5.4.4 STC Bypass / ATE Interface

This is a production test interface. This section bypasses the self-test FSM. The OPMISR signal interface is brought out directly to the module ports, and these are accessible to the ATE (tester) at the device level using the test controller module. The intent of the block is to provide capability for fault isolation for parts failing the logic self test run.

This block receives two sets of signals; one from the device test controller, and another similar set from the self-test FSM (test sequencer). The bypass indicator signal is used to select one of the two sets of signals to be routed to the OPMISR controller.

28.5.4.5 VBUSP Interface

The control and the status registers of the STC module can be accessed through the VBUSP interface. During application programming, configuration registers are programmed through the peripheral interface, to enable and run the self-test controller.

28.5.5 STC Flow

Figure 28-142. STC Flow (1 of 2)

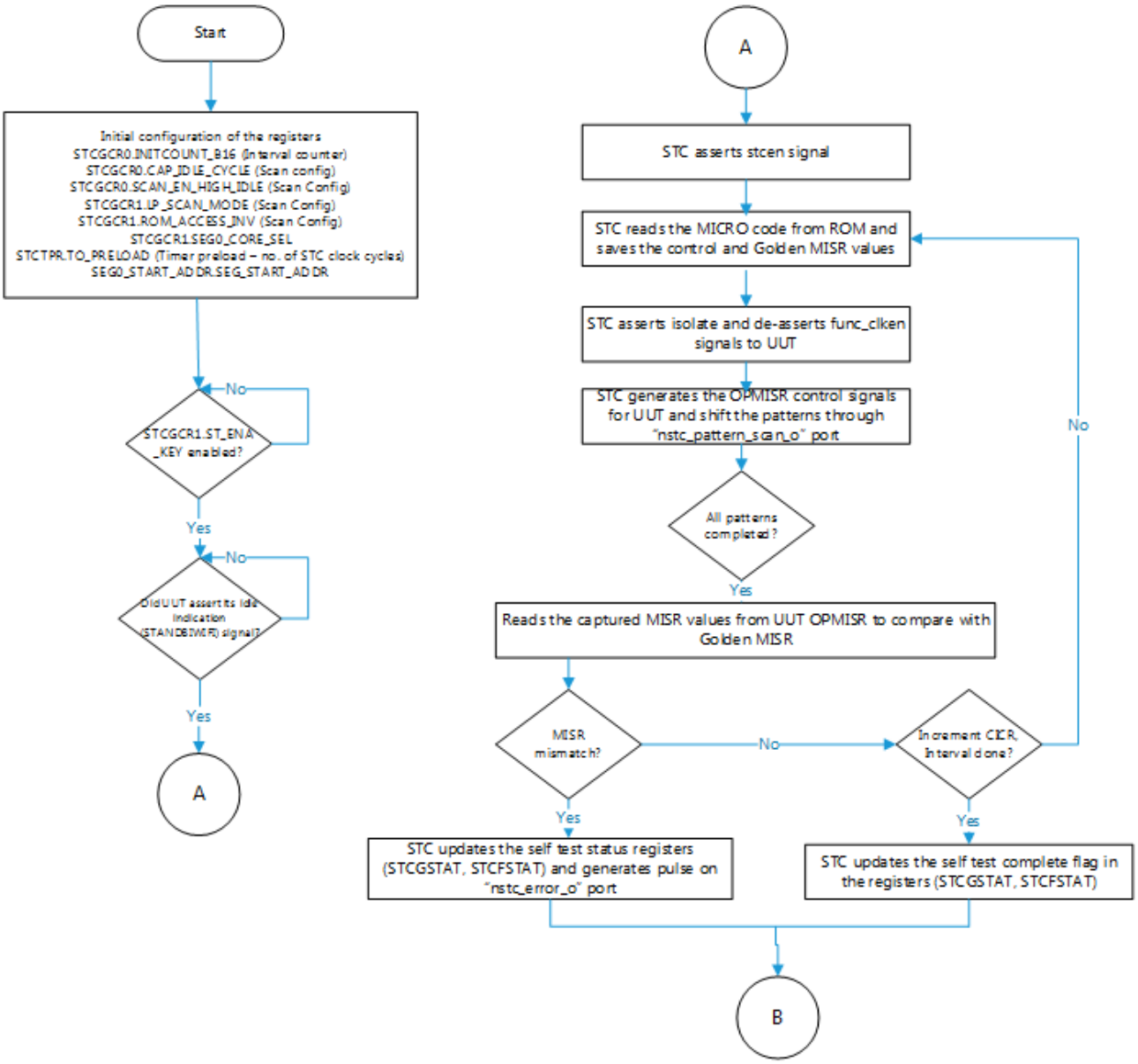
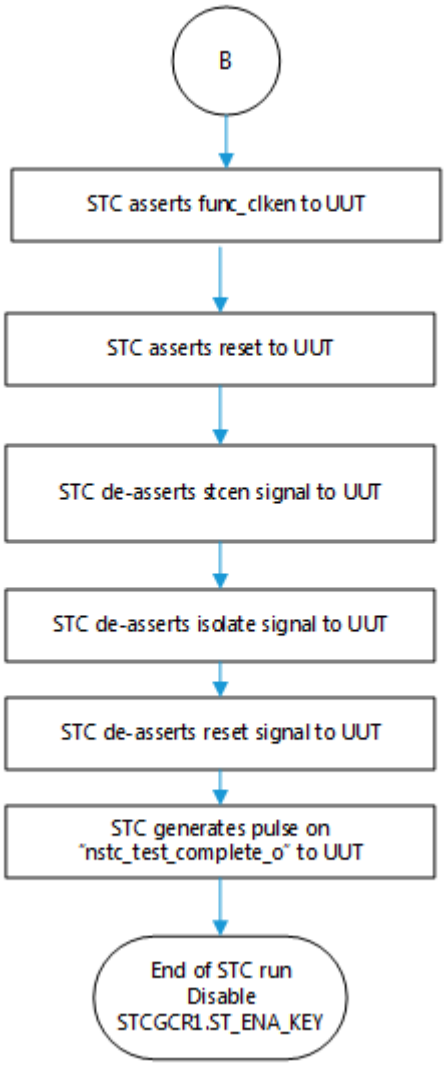


Figure 28-143. STC Flow (2 of 2)



28.5.6 ROM Organization

Table 28-131. ROM Organization for 2 Intervals

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
INTERVAL 0									
CFG for interval 0, when rom_access_inversion =0	Reserved	pattern_count[9:0]	Reserved	clk_domain_off[7:0]	Reserved	Seg_ID[1]	Seg_ID[0]	FT	TR_T

Table 28-131. ROM Organization for 2 Intervals (continued)

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
MISR for interval 0, when rom_access_inversion =0	MISR_GOLDEN[895:840]								
	MISR_GOLDEN[839:784]								
	MISR_GOLDEN[783:728]								
	MISR_GOLDEN[727:672]								
	MISR_GOLDEN[671:616]								
	MISR_GOLDEN[615:560]								
	MISR_GOLDEN[559:504]								
	MISR_GOLDEN[503:448]								
	MISR_GOLDEN[447:392]								
	MISR_GOLDEN[391:336]								
	MISR_GOLDEN[335:280]								
	MISR_GOLDEN[279:224]								
	MISR_GOLDEN[223:168]								
	MISR_GOLDEN[167:112]								
	MISR_GOLDEN[111:56]								
	MISR_GOLDEN[55:0]								
LP_MISR for interval 0, when rom_access_inversion =0	LP_MISR_GOLDEN[895:840]								
	LP_MISR_GOLDEN[839:784]								
	LP_MISR_GOLDEN[783:728]								
	LP_MISR_GOLDEN[727:672]								
	LP_MISR_GOLDEN[671:616]								
	LP_MISR_GOLDEN[615:560]								
	LP_MISR_GOLDEN[559:504]								
	LP_MISR_GOLDEN[503:448]								
	LP_MISR_GOLDEN[447:392]								
	LP_MISR_GOLDEN[391:336]								
	LP_MISR_GOLDEN[335:280]								
	LP_MISR_GOLDEN[279:224]								
	LP_MISR_GOLDEN[223:168]								
	LP_MISR_GOLDEN[167:112]								
	LP_MISR_GOLDEN[111:56]								
	LP_MISR_GOLDEN[55:0]								
Patterns for interval 0	P1_SD8[6:0]	P1_SD7[6:0]	P1_SD6[6:0]	….	….	…	P1_SD1[6:0]		
		…	…	…	…	…	…	P1_SD9[6:0]	
		…	…	…	…	…	…	…	…

Table 28-131. ROM Organization for 2 Intervals (continued)

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
LP_MISR for interval 0, when rom_access_inversion =1	LP_INV_MISR_GOLDEN[55:0]								
	LP_INV_MISR_GOLDEN[111:56]								
	LP_INV_MISR_GOLDEN[167:112]								
	LP_INV_MISR_GOLDEN[223:168]								
	LP_INV_MISR_GOLDEN[279:224]								
	LP_INV_MISR_GOLDEN[335:280]								
	LP_INV_MISR_GOLDEN[391:336]								
	LP_INV_MISR_GOLDEN[447:392]								
	LP_INV_MISR_GOLDEN[503:448]								
	LP_INV_MISR_GOLDEN[559:504]								
	LP_INV_MISR_GOLDEN[615:560]								
	LP_INV_MISR_GOLDEN[671:616]								
	LP_INV_MISR_GOLDEN[727:672]								
	LP_INV_MISR_GOLDEN[783:728]								
	LP_INV_MISR_GOLDEN[839:784]								
	LP_INV_MISR_GOLDEN[895:840]								
MISR for interval 0, when rom_access_inversion =1	INV_MISR_GOLDEN[55:0]								
	INV_MISR_GOLDEN[111:56]								
	INV_MISR_GOLDEN[167:112]								
	INV_MISR_GOLDEN[223:168]								
	INV_MISR_GOLDEN[279:224]								
	INV_MISR_GOLDEN[335:280]								
	INV_MISR_GOLDEN[391:336]								
	INV_MISR_GOLDEN[447:392]								
	INV_MISR_GOLDEN[503:448]								
	INV_MISR_GOLDEN[559:504]								
	INV_MISR_GOLDEN[615:560]								
	INV_MISR_GOLDEN[671:616]								
	INV_MISR_GOLDEN[727:672]								
	INV_MISR_GOLDEN[783:728]								
	INV_MISR_GOLDEN[839:784]								
	INV_MISR_GOLDEN[895:840]								
CFG for interval 0, when rom_access_inversion =1 (same as when_rom_access_inversion =0)	Reserved	pattern_count[9:0]	Reserved	clk_domain_off[7:0]	Reserved	Seg_ID[1]	Seg_ID[0]	FT	TR_T
INTERVAL 1									
CFG for interval 1, when rom_access_inversion =0	Reserved	pattern_count[9:0]	Reserved	clk_domain_off[7:0]	Reserved	Seg_ID[1]	Seg_ID[0]	FT	TR_T

Table 28-131. ROM Organization for 2 Intervals (continued)

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
MISR for interval 1, when rom_access_inversion =0	MISR_GOLDEN[895:840]								
	MISR_GOLDEN[839:784]								
	MISR_GOLDEN[783:728]								
	MISR_GOLDEN[727:672]								
	MISR_GOLDEN[671:616]								
	MISR_GOLDEN[615:560]								
	MISR_GOLDEN[559:504]								
	MISR_GOLDEN[503:448]								
	MISR_GOLDEN[447:392]								
	MISR_GOLDEN[391:336]								
	MISR_GOLDEN[335:280]								
	MISR_GOLDEN[279:224]								
	MISR_GOLDEN[223:168]								
	MISR_GOLDEN[167:112]								
	MISR_GOLDEN[111:56]								
	MISR_GOLDEN[55:0]								
LP_MISR for interval 1, when rom_access_inversion =0	LP_MISR_GOLDEN[895:840]								
	LP_MISR_GOLDEN[839:784]								
	LP_MISR_GOLDEN[783:728]								
	LP_MISR_GOLDEN[727:672]								
	LP_MISR_GOLDEN[671:616]								
	LP_MISR_GOLDEN[615:560]								
	LP_MISR_GOLDEN[559:504]								
	LP_MISR_GOLDEN[503:448]								
	LP_MISR_GOLDEN[447:392]								
	LP_MISR_GOLDEN[391:336]								
	LP_MISR_GOLDEN[335:280]								
	LP_MISR_GOLDEN[279:224]								
	LP_MISR_GOLDEN[223:168]								
	LP_MISR_GOLDEN[167:112]								
	LP_MISR_GOLDEN[111:56]								
	LP_MISR_GOLDEN[55:0]								
Patterns for interval 1	P1_SD8[6:0]	P1_SD7[6:0]	P1_SD6[6:0]	….	….	…	P1_SD1[6:0]		
		…	…	…	…	…	…	P1_SD9[6:0]	
		…	…	…	…	…	…	…	…

Table 28-131. ROM Organization for 2 Intervals (continued)

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
LP_MISR for interval 1, when rom_access_inversion =1	LP_INV_MISR_GOLDEN[55:0]								
	LP_INV_MISR_GOLDEN[111:56]								
	LP_INV_MISR_GOLDEN[167:112]								
	LP_INV_MISR_GOLDEN[223:168]								
	LP_INV_MISR_GOLDEN[279:224]								
	LP_INV_MISR_GOLDEN[335:280]								
	LP_INV_MISR_GOLDEN[391:336]								
	LP_INV_MISR_GOLDEN[447:392]								
	LP_INV_MISR_GOLDEN[503:448]								
	LP_INV_MISR_GOLDEN[559:504]								
	LP_INV_MISR_GOLDEN[615:560]								
	LP_INV_MISR_GOLDEN[671:616]								
	LP_INV_MISR_GOLDEN[727:672]								
	LP_INV_MISR_GOLDEN[783:728]								
	LP_INV_MISR_GOLDEN[839:784]								
	LP_INV_MISR_GOLDEN[895:840]								
MISR for interval 1, when rom_access_inversion =1	INV_MISR_GOLDEN[55:0]								
	INV_MISR_GOLDEN[111:56]								
	INV_MISR_GOLDEN[167:112]								
	INV_MISR_GOLDEN[223:168]								
	INV_MISR_GOLDEN[279:224]								
	INV_MISR_GOLDEN[335:280]								
	INV_MISR_GOLDEN[391:336]								
	INV_MISR_GOLDEN[447:392]								
	INV_MISR_GOLDEN[503:448]								
	INV_MISR_GOLDEN[559:504]								
	INV_MISR_GOLDEN[615:560]								
	INV_MISR_GOLDEN[671:616]								
	INV_MISR_GOLDEN[727:672]								
	INV_MISR_GOLDEN[783:728]								
	INV_MISR_GOLDEN[839:784]								
	INV_MISR_GOLDEN[895:840]								
CFG for interval 1, when rom_access_inversion =1 (same as when_rom_access_inversion =0)	Reserved	pattern_count[9:0]	Reserved	clk_domain_off[7:0]	Reserved	Seg_ID[1]	Seg_ID[0]	FT	TR_T

The ROM contains the data to be processed by STC for the self-test run. This includes the control fields such as Segment ID, Pattern Count, and Golden MISR value for the STC, and the pattern scan data for the OPMISR controller.

The ROM space is divided into chunks, with each chunk containing the data corresponding to one OPMISR interval. The size required for an interval varies depending on the number patterns packed into the interval and the length of internal scan chains required.

Because each interval requires 64 rows of ROM for storing control and Golden MISR values, minimizing the number of intervals by packing more patterns into each interval provides the best ROM size. This works best if the self-test must be run only as a part of the boot-up sequence. However, if the self-test is performed during application IDLE time, the number of patterns that can be packed into each interval will be dictated by the IDLE time available for the self-test, because an interval is the smallest granularity of a self-test run.

Details of the ROM image micro-code fields are given in the following sections.

28.5.6.1 TR_T: Transition Delay Methodology Type

This specifies the transition delay methodology for the current transition delay interval.

0	Launch-on-System-Clock
1	Launch-on-Last-Shift

28.5.6.2 FT: Fault Model for the BIST Run

This specifies the fault model for the current interval of the test.

0	Stuck-at
1	Transition Delay

28.5.6.3 SEG_ID[1:0]

This indicates which logical segment is selected for the associated interval during the self-test run.

SEG_SEL[1:0]	Segment Under Test
00	Segment 0
01	Segment 1
10	Segment 2
11	Segment 3

28.5.6.4 Pattern Count (patt_count[9:0])

This specifies the number of scan data patterns within a self-test interval. The pattern counts can vary from a minimum of 2 to a maximum of 1024.

patt_count[9:0]	Patterns per Interval
00_0000_0000	Not a valid interval [defaults to 2 patterns per interval]
00_0000_0001	2 patterns per interval
00_0000_0010	3 patterns per interval
...	...
11_1111_1110	1023 patterns per interval
11_1111_1111	1024 patterns per interval

28.5.6.5 CLK_DOMAIN_OFF[7:0] (Not Supported in This Version)

This specifies whether to enable or switch off the particular clock domain or clock groups (consisting of multiple clock domains of the same maximum spec frequency) of the BIST'ed CORE logic during the capture phase of the At-speed OPMISR pattern of the current interval.

Domain	Value	Description
clk_domain_off[0]	0	Clock0 enabled
	1	Clock0 disabled
clk_domain_off[1]	0	Clock1 enabled
	1	Clock1 disabled
clk_domain_off[2]	0	Clock2 enabled
	1	Clock2 disabled

Domain	Value	Description
clk_domain_off[3]	0	Clock3 enabled
	1	Clock3 disabled
clk_domain_off[4]	0	Clock4 enabled
	1	Clock4 disabled
clk_domain_off[5]	0	Clock5 enabled
	1	Clock5 disabled
clk_domain_off[6]	0	Clock6 enabled
	1	Clock6 disabled
clk_domain_off[7]	0	Clock7 enabled
	1	Clock7 disabled

28.5.6.6 MISR_GOLDEN[895:0]: Golden Signature Data Bits

This part of ROM contains the golden signature data of the current interval. This value is used to compare with the actual MISR value, when ST_GCR1.ROM_ACCESS_INV=0 and ST_GCR1.LP_SCAN_MODE=0, to generate the pass/fail information of the interval.

28.5.6.7 LP_MISR_GOLDEN[895:0]: Low Power Mode Golden Signature Data Bits

This part of ROM contains the LP golden signature data of the current interval. This value is used to compare with the actual MISR value, when STCGCR1.ROM_ACCESS_INV=0 and STCGCR1.LP_SCAN_MODE=1, to generate the pass/fail information of the interval.

28.5.6.8 INV_MISR_GOLDEN[895:0]: Inverse Mode Golden Signature Data Bits

This part of ROM contains the inverse mode golden signature data of the current interval. This value is used to compare with the actual MISR value, when STCGCR1.ROM_ACCESS_INV=1 and STCGCR1.LP_SCAN_MODE=0, to generate the pass/fail information of the interval.

28.5.6.9 LP_INV_MISR_GOLDEN[895:0]: Low Power Inverse Mode Golden Signature Data Bits

This part of ROM contains the low-power inverse mode golden signature data of the current interval. This value is used to compare with the actual MISR value, when STCGCR1.ROM_ACCESS_INV=1 and STCGCR1.LP_SCAN_MODE=1, to generate the pass/fail information of the interval.

28.5.6.10 Pn_SDm[7:0] (n - no. of patterns, m - scan chain length): OP-MISR Scan Data

This part of the ROM contains the scan data corresponding to each pattern. Each interval can have n number of scan patterns, as defined in the patt_count field. The number of 7bits of scan data in a pattern is equal to the length of the scan chain formed inside the UUT.

28.5.7 STC Registers

Table 28-132 lists the memory-mapped registers for the STC. All register offset addresses not listed in Table 28-132 should be considered as reserved locations and the register contents should not be modified.

Table 28-132. STC Registers

Offset	Acronym	Register Name	Section
0h	STCGCR0	Self test Global control Reg0	Section 28.5.7.1
4h	STCGCR1	Self test Global control Reg1	Section 28.5.7.2
8h	STCTPR	Time out counter preload register	Section 28.5.7.3
Ch	STC_CADDR	Current Address register for CORE1	Section 28.5.7.4
10h	STCCICR	Current Interval count register	Section 28.5.7.5
14h	STCGSTAT	Global Status Register	Section 28.5.7.6
18h	STCFSTAT	Fail Status Register	Section 28.5.7.7
1Ch	STCSCSCR	Signature compare Self Check Register	Section 28.5.7.8
20h	STC_CADDR2	Current Address register for CORE2	Section 28.5.7.9
24h	STC_CLKDIV	Clock Divider Register	Section 28.5.7.10
28h	STC_SEGPLR	Segment 1st interval Preload Register	Section 28.5.7.11
2Ch	SEG0_START_ADDR	ROM Start address for Segment0	Section 28.5.7.12
30h	SEG1_START_ADDR	ROM Start address for Segment1	Section 28.5.7.13
34h	SEG2_START_ADDR	ROM Start address for Segment2	Section 28.5.7.14
38h	SEG3_START_ADDR	ROM Start address for Segment3	Section 28.5.7.15
3Ch	CORE1_CURMISR_0	Holds the MISR signature for CORE1	Section 28.5.7.16
40h	CORE1_CURMISR_1	Holds the MISR signature for CORE1	Section 28.5.7.17
44h	CORE1_CURMISR_2	Holds the MISR signature for CORE1	Section 28.5.7.18
48h	CORE1_CURMISR_3	Holds the MISR signature for CORE1	Section 28.5.7.19
4Ch	CORE1_CURMISR_4	Holds the MISR signature for CORE1	Section 28.5.7.20
50h	CORE1_CURMISR_5	Holds the MISR signature for CORE1	Section 28.5.7.21
54h	CORE1_CURMISR_6	Holds the MISR signature for CORE1	Section 28.5.7.22
58h	CORE1_CURMISR_7	Holds the MISR signature for CORE1	Section 28.5.7.23
5Ch	CORE1_CURMISR_8	Holds the MISR signature for CORE1	Section 28.5.7.24
60h	CORE1_CURMISR_9	Holds the MISR signature for CORE1	Section 28.5.7.25
64h	CORE1_CURMISR_10	Holds the MISR signature for CORE1	Section 28.5.7.26
68h	CORE1_CURMISR_11	Holds the MISR signature for CORE1	Section 28.5.7.27
6Ch	CORE1_CURMISR_12	Holds the MISR signature for CORE1	Section 28.5.7.28
70h	CORE1_CURMISR_13	Holds the MISR signature for CORE1	Section 28.5.7.29
74h	CORE1_CURMISR_14	Holds the MISR signature for CORE1	Section 28.5.7.30
78h	CORE1_CURMISR_15	Holds the MISR signature for CORE1	Section 28.5.7.31
7Ch	CORE1_CURMISR_16	Holds the MISR signature for CORE1	Section 28.5.7.32
80h	CORE1_CURMISR_17	Holds the MISR signature for CORE1	Section 28.5.7.33
84h	CORE1_CURMISR_18	Holds the MISR signature for CORE1	Section 28.5.7.34
88h	CORE1_CURMISR_19	Holds the MISR signature for CORE1	Section 28.5.7.35
8Ch	CORE1_CURMISR_20	Holds the MISR signature for CORE1	Section 28.5.7.36
90h	CORE1_CURMISR_21	Holds the MISR signature for CORE1	Section 28.5.7.37
94h	CORE1_CURMISR_22	Holds the MISR signature for CORE1	Section 28.5.7.38
98h	CORE1_CURMISR_23	Holds the MISR signature for CORE1	Section 28.5.7.39
9Ch	CORE1_CURMISR_24	Holds the MISR signature for CORE1	Section 28.5.7.40
A0h	CORE1_CURMISR_25	Holds the MISR signature for CORE1	Section 28.5.7.41
A4h	CORE1_CURMISR_26	Holds the MISR signature for CORE1	Section 28.5.7.42
A8h	CORE1_CURMISR_27	Holds the MISR signature for CORE1	Section 28.5.7.43

Table 28-132. STC Registers (continued)

Offset	Acronym	Register Name	Section
ACh	CORE2_CURMISR_0	Holds the MISR signature for CORE2	Section 28.5.7.44
B0h	CORE2_CURMISR_1	Holds the MISR signature for CORE2	Section 28.5.7.45
B4h	CORE2_CURMISR_2	Holds the MISR signature for CORE2	Section 28.5.7.46
B8h	CORE2_CURMISR_3	Holds the MISR signature for CORE2	Section 28.5.7.47
BCh	CORE2_CURMISR_4	Holds the MISR signature for CORE2	Section 28.5.7.48
C0h	CORE2_CURMISR_5	Holds the MISR signature for CORE2	Section 28.5.7.49
C4h	CORE2_CURMISR_6	Holds the MISR signature for CORE2	Section 28.5.7.50
C8h	CORE2_CURMISR_7	Holds the MISR signature for CORE2	Section 28.5.7.51
CCh	CORE2_CURMISR_8	Holds the MISR signature for CORE2	Section 28.5.7.52
D0h	CORE2_CURMISR_9	Holds the MISR signature for CORE2	Section 28.5.7.53
D4h	CORE2_CURMISR_10	Holds the MISR signature for CORE2	Section 28.5.7.54
D8h	CORE2_CURMISR_11	Holds the MISR signature for CORE2	Section 28.5.7.55
DCh	CORE2_CURMISR_12	Holds the MISR signature for CORE2	Section 28.5.7.56
E0h	CORE2_CURMISR_13	Holds the MISR signature for CORE2	Section 28.5.7.57
E4h	CORE2_CURMISR_14	Holds the MISR signature for CORE2	Section 28.5.7.58
E8h	CORE2_CURMISR_15	Holds the MISR signature for CORE2	Section 28.5.7.59
ECh	CORE2_CURMISR_16	Holds the MISR signature for CORE2	Section 28.5.7.60
F0h	CORE2_CURMISR_17	Holds the MISR signature for CORE2	Section 28.5.7.61
F4h	CORE2_CURMISR_18	Holds the MISR signature for CORE2	Section 28.5.7.62
F8h	CORE2_CURMISR_19	Holds the MISR signature for CORE2	Section 28.5.7.63
FCh	CORE2_CURMISR_20	Holds the MISR signature for CORE2	Section 28.5.7.64
100h	CORE2_CURMISR_21	Holds the MISR signature for CORE2	Section 28.5.7.65
104h	CORE2_CURMISR_22	Holds the MISR signature for CORE2	Section 28.5.7.66
108h	CORE2_CURMISR_23	Holds the MISR signature for CORE2	Section 28.5.7.67
10Ch	CORE2_CURMISR_24	Holds the MISR signature for CORE2	Section 28.5.7.68
110h	CORE2_CURMISR_25	Holds the MISR signature for CORE2	Section 28.5.7.69
114h	CORE2_CURMISR_26	Holds the MISR signature for CORE2	Section 28.5.7.70
118h	CORE2_CURMISR_27	Holds the MISR signature for CORE2	Section 28.5.7.71

Complex bit access types are encoded to fit into small table cells. Table 28-133 shows the codes that are used for access types in this section.

Table 28-133. STC Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

28.5.7.1 STCGCR0 Register (Offset = 0h) [reset = 00010120h]

STCGCR0 is shown in Figure 28-144 and described in Table 28-134.

Return to Summary Table.

Self test Global control Reg0. *NOT BYTE ACCESSIBLE

Figure 28-144. STCGCR0 Register

31	30	29	28	27	26	25	24
INTCOUNT_B16							
R/W-1h							
23	22	21	20	19	18	17	16
INTCOUNT_B16							
R/W-1h							
15	14	13	12	11	10	9	8
NU0					CAP_IDLE_CYCLE		
R-0h					R/W-1h		
7	6	5	4	3	2	1	0
SCANEN_HIGH_CAP_IDLE_CYCLE			NU1			RS_CNT_B1	
R/W-1h			R-0h			R/W-0h	

Table 28-134. STCGCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	INTCOUNT_B16	R/W	1h	Number of intervals of the self test run (RWP - Read, Priviledge Mode Write only) Count of intervals that need to be covered for a specific selftest run. The selftest controller sends out “complete” indication once it runs all of the intervals programmed in this field. INTCOUNT_B16=0 is an invalid configuration for a selftest.
15-11	NU0	R	0h	Reserved bits
10-8	CAP_IDLE_CYCLE	R/W	1h	Idle cycles before and after capture clock (RWP - Read, Priviledge Mode Write only) Idle Cycles before and after capture clock. This value is used to insert that many idle cycles in the Capture phase. Programmable idle cycles allow implementation flexibility on SCAN_EN signal at chip level based on the size of the UUT and timing requirements.
7-5	SCANEN_HIGH_CAP_IDLE_CYCLE	R/W	1h	Idle cycles before and after capture clock (RWP - Read, Priviledge Mode Write only). *NOT BYTE ACCESSIBLE Idle Cycles between scan_en going high to func_clk_en generation and scan_en going high to misr_log_en generation. This value is used to insert that many idle cycles in the shift clock (scan_en going high to func_clk_en generation) and misr_log_clk (scan_en going high to misr_log_en generation) generation. Programmable idle cycles allow implementation flexibility on SCAN_EN signal at chip level based on the size of the UUT and timing requirements.
4-2	NU1	R	0h	Reserved bits
1-0	RS_CNT_B1	R/W	0h	Restart/Continue or preload (RWP - Read, Priviledge Mode Write only) This bit specifies the selftest controller whether to continue the run from next interval onwards, restart from ROM address 0 or preload from a prescribed interval. This bit gets reset after the completion of selftest run. 00 = Continue NSTC run from previous interval 01 = Restart NSTC run from ROM address 0 1X = Start from segment number specified in STC_SEGPLR register

28.5.7.2 STCGCR1 Register (Offset = 4h) [reset = 25h]

STCGCR1 is shown in Figure 28-145 and described in Table 28-135.

Return to Summary Table.

Self test Global control Reg1

Figure 28-145. STCGCR1 Register

31	30	29	28	27	26	25	24
NU2							
R-0h							
23	22	21	20	19	18	17	16
NU2							
R-0h							
15	14	13	12	11	10	9	8
NU2				SEG0_CORE_SEL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
NU3	CODEC_SPREAD_MODE	LP_SCAN_MODE	ROM_ACCESS_INV	ST_ENA_B4			
R-0h	R/W-0h	R/W-1h	R/W-0h	R/W-5h			

Table 28-135. STCGCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU2	R	0h	Reserved bits
11-8	SEG0_CORE_SEL	R/W	0h	Selects the Segment0 CORE for self test (RWP - Read, Priviledge Mode Write only) Select the Segment0 CORE for Self -Test 0001 = Select CORE for selftest Other = CORE not selected.
7	NU3	R	0h	Reserved bits
6	CODEC_SPREAD_MODE	R/W	0h	Codec Spread Mode control signal (RWP - Read, Priviledge Mode Write only) This bit is used to configure the codec in spread / X-OR mode. 1 = Spread mode 0 = XOR mode
5	LP_SCAN_MODE	R/W	1h	LP scan mode (RWP - Read, Priviledge Mode Write only) This bit is used to decide the scan configuration: 1 = Operates in Low Power Scan Mode. 0 = Operates in Normal Scan Mode.
4	ROM_ACCESS_INV	R/W	0h	Rom access inversion mode (RWP - Read, Priviledge Mode Write only) - NOT SUPPORTED
3-0	ST_ENA_B4	R/W	5h	Self test enable key (RWP - Read, Priviledge Mode Write only) 1010 = Self test run enabled All values other than 1010 = Self test run disabled

28.5.7.3 STCTPR Register (Offset = 8h) [reset = FFFFFFFFh]

STCTPR is shown in Figure 28-146 and described in Table 28-136.

Return to Summary Table.

Time out counter preload register

Figure 28-146. STCTPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO_PRELOAD																															
R/W-FFFFFFFFh																															

Table 28-136. STCTPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TO_PRELOAD	R/W	FFFFFFFFh	Self test time out preload (RWP - Read, Priviledge Mode Write only) This register contains the total number of STC clock cycles it will take before a self-test timeout error will be triggered after the initiation of the self-test run. This is a fail safe feature to avoid system hang-up situation on account of any run away self test issues. This register should be loaded with a meaningful count value for this feature to be effective. This register value (preload count value) gets loaded into the self test timeout down counter whenever a self test run is initiated (ST_ENA is enabled). and gets disabled on completion of a self test run.

28.5.7.4 STC_CADDR Register (Offset = Ch) [reset = 0h]

STC_CADDR is shown in Figure 28-147 and described in Table 28-137.

Return to Summary Table.

Current Address register for CORE1

Figure 28-147. STC_CADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
R-0h																															

Table 28-137. STC_CADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Current ROM Address for CORE1 This register reflects the current ROM address (for micro code load) accessed during selftest for CORE1 in of case segment0 and all the remaining segmentsn where n = 1 to 3).

28.5.7.5 STCCICR Register (Offset = 10h) [reset = 0h]

STCCICR is shown in Figure 28-148 and described in Table 28-138.

Return to Summary Table.

Current Interval count register

Figure 28-148. STCCICR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE2_ICOUNT																CORE1_ICOUNT															
R-0h																R-0h															

Table 28-138. STCCICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CORE2_ICOUNT	R	0h	Specifies the last interval number for CORE2 This specifies the Last executed Interval number for CORE2 of Segment0 if self test is being executed for secondary core as well. This field is applicable only for Segment 0.
15-0	CORE1_ICOUNT	R	0h	Specifies the last interval number for CORE1 This specifies the Last executed Interval number of a self-test run.

28.5.7.6 STCGSTAT Register (Offset = 14h) [reset = 0h]

STCGSTAT is shown in Figure 28-149 and described in Table 28-139.

Return to Summary Table.

Global Status Register

Figure 28-149. STCGSTAT Register

31	30	29	28	27	26	25	24
NU4							
R-0h							
23	22	21	20	19	18	17	16
NU4							
R-0h							
15	14	13	12	11	10	9	8
NU4				ST_ACTIVE			
R-0h				R-0h			
7	6	5	4	3	2	1	0
NU5						TEST_FAIL	TEST_DONE
R-0h						R-0h	R-0h

Table 28-139. STCGSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU4	R	0h	Reserved bits
11-8	ST_ACTIVE	R	0h	Tells whether self test is currently active or not. 1010 = Self test is active Others = SelfTest is not active Once the self-test completes and ST_ENA_B4 key is cleared, this field will reflect the inactive value.
7-2	NU5	R	0h	Reserved bits
1	TEST_FAIL	R	0h	Test_fail flag (RCP - Read, Clear on Writing in Priviledge Mode) 0 = Self test run has not failed 1 = SelfTest run has failed. Write Clear.
0	TEST_DONE	R	0h	Test_done_flag (RCP - Read, Clear on Writing in Priviledge Mode) 0 = Not completed 1 = SelfTest run Completed

28.5.7.7 STCFSTAT Register (Offset = 18h) [reset = 0h]

STCFSTAT is shown in Figure 28-150 and described in Table 28-140.

Return to Summary Table.

Fail Status Register

Figure 28-150. STCFSTAT Register

31	30	29	28	27	26	25	24
NU6							
R-0h							
23	22	21	20	19	18	17	16
NU6							
R-0h							
15	14	13	12	11	10	9	8
NU6							
R-0h							
7	6	5	4	3	2	1	0
NU6			FSEG_ID		TO_ER_B1	CPU2_FAIL_B1	CPU1_FAIL_B1
R-0h			R-0h		R-0h	R-0h	R-0h

Table 28-140. STCFSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	NU6	R	0h	Reserved bits
4-3	FSEG_ID	R	0h	Failed Segment ID (RCP - Read, Clear on Writing in Priviledge Mode) This field captures the Segment number for which any of the failures like TO_ER_B1, CPU1_FAIL_B1 and CPU2_FAIL_B1 occur. 00 = Failure on Segment 0 01 = Failure on Segment 1 10 = Failure on Segment 2 11 = Failure on Segment 3
2	TO_ER_B1	R	0h	Tells whether self test failed because of time out error (RCP - Read, Clear on Writing in Priviledge Mode) 0 = No time out error occurred 1 = SelfTest run failed due to a timeout error
1	CPU2_FAIL_B1	R	0h	Tells whether MISR mismatch happenned in CORE2 when in Segment0 mode (RCP - Read, Clear on Writing in Priviledge Mode) 0 = No MISR mismatch for CORE2 1 = Self test run failed due to MISR mismatch for CORE2
0	CPU1_FAIL_B1	R	0h	Tells whether MISR mismatch happenned in CORE1 (RCP - Read, Clear on Writing in Priviledge Mode) Applicable to all segments. 0 = No MISR mismatch for CORE1 1 = Self test run failed due to MISR mismatch for CORE1

28.5.7.8 STCSCSCR Register (Offset = 1Ch) [reset = 0h]

STCSCSCR is shown in Figure 28-151 and described in Table 28-141.

Return to Summary Table.

Signature compare Self Check Register

Figure 28-151. STCSCSCR Register

31	30	29	28	27	26	25	24
NU7							
R-0h							
23	22	21	20	19	18	17	16
NU7							
R-0h							
15	14	13	12	11	10	9	8
NU7							
R-0h							
7	6	5	4	3	2	1	0
NU7			FAULT_INS_B1	SELF_CHECK_KEY_B4			
R-0h			R/W-0h	R/W-0h			

Table 28-141. STCSCSCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	NU7	R	0h	Reserved bits
4	FAULT_INS_B1	R/W	0h	Fault Insertion bit (RWP - Read, Priviledge Mode Write only) 0 = No fault insertion. 1 = Inserts fault in the logic unedr test which will make signature compare fail. This feature is used as diagnostic check of the STC IP.
3-0	SELF_CHECK_KEY_B4	R/W	0h	Signature compare logic self check key enable/disable (RWP - Read, Priviledge Mode Write only) 1010 = Signature compare logic Self Check is enabled All values other than 1010 = Signature compare logic Self Check is disabled

28.5.7.9 STC_CADDR2 Register (Offset = 20h) [reset = 0h]

STC_CADDR2 is shown in Figure 28-152 and described in Table 28-142.

Return to Summary Table.

Current Address register for CORE2

Figure 28-152. STC_CADDR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
R-0h																															

Table 28-142. STC_CADDR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Current ROM Address for CORE2 This register reflects the current ROM address(for micro code load) accessed during selftest for CORE2 in of case segment0.

28.5.7.10 STC_CLKDIV Register (Offset = 24h) [reset = 0h]

STC_CLKDIV is shown in Figure 28-153 and described in Table 28-143.

Return to Summary Table.

Clock Divider Register

Figure 28-153. STC_CLKDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU8					CLKDIV0			NU9					CLKDIV1		
R-0h					R/W-0h			R-0h					R/W-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU10					CLKDIV2			NU11					CLKDIV3		
R-0h					R/W-0h			R-0h					R/W-0h		

Table 28-143. STC_CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	NU8	R	0h	Reserved bits
26-24	CLKDIV0	R/W	0h	Clock division for Seg0 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 0
23-19	NU9	R	0h	Reserved bits
18-16	CLKDIV1	R/W	0h	Clock division for Seg1 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 1
15-11	NU10	R	0h	Reserved bits
10-8	CLKDIV2	R/W	0h	Clock division for Seg2 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 2
7-3	NU11	R	0h	Reserved bits
2-0	CLKDIV3	R/W	0h	Clock division for Seg3 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 3

28.5.7.11 STC_SEGPLR Register (Offset = 28h) [reset = 0h]

STC_SEGPLR is shown in Figure 28-154 and described in Table 28-144.

Return to Summary Table.

Segment 1st interval Preload Register

Figure 28-154. STC_SEGPLR Register

31	30	29	28	27	26	25	24
NU12							
R-0h							
23	22	21	20	19	18	17	16
NU12							
R-0h							
15	14	13	12	11	10	9	8
NU12							
R-0h							
7	6	5	4	3	2	1	0
NU12						SEGID_PLOAD	
R-0h						R/W-0h	

Table 28-144. STC_SEGPLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU12	R	0h	Reserved bits
1-0	SEGID_PLOAD	R/W	0h	Segment number for which preload is to be started (RWP - Read, Priviledge Mode Write only) This specifies the segment for which the address of its First interval will be pre-loaded into the NSTC ROM address counter. The 1st address of each segment are defined in SEGx_START_ADDR register.The address of the 1st interval of the selected segment is loaded into the NSTC ROM address counter when the RS_CNT_B1 bits of STC_GCR0 are set to 1X 00 = Preload the address of the 1st interval of segment 0. 01 = Preload the address of the 1st interval of segment 1. 10 = Preload the address of the 1st interval of segment 2. 11 = Preload the address of the 1st interval of segment 3.

28.5.7.12 SEG0_START_ADDR Register (Offset = 2Ch) [reset = 0h]

SEG0_START_ADDR is shown in Figure 28-155 and described in Table 28-145.

Return to Summary Table.

ROM Start address for Segment0

Figure 28-155. SEG0_START_ADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU13												SEG_START_ADDR																			
R-0h												R/W-0h																			

Table 28-145. SEG0_START_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU13	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 0 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) “PRELOAD” option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register.

28.5.7.13 SEG1_START_ADDR Register (Offset = 30h) [reset = 0h]

SEG1_START_ADDR is shown in Figure 28-156 and described in Table 28-146.

Return to Summary Table.

ROM Start address for Segment1

Figure 28-156. SEG1_START_ADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU14												SEG_START_ADDR																			
R-0h												R/W-0h																			

Table 28-146. SEG1_START_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU14	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 1 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) “PRELOAD” option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register.

28.5.7.14 SEG2_START_ADDR Register (Offset = 34h) [reset = 0h]

SEG2_START_ADDR is shown in Figure 28-157 and described in Table 28-147.

Return to Summary Table.

ROM Start address for Segment2

Figure 28-157. SEG2_START_ADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU15												SEG_START_ADDR																			
R-0h												R/W-0h																			

Table 28-147. SEG2_START_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU15	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 2 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) “PRELOAD” option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register.

28.5.7.15 SEG3_START_ADDR Register (Offset = 38h) [reset = 0h]

SEG3_START_ADDR is shown in Figure 28-158 and described in Table 28-148.

Return to Summary Table.

ROM Start address for Segment3

Figure 28-158. SEG3_START_ADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU16												SEG_START_ADDR																			
R-0h												R/W-0h																			

Table 28-148. SEG3_START_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU16	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 3 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) “PRELOAD” option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register.

28.5.7.16 CORE1_CURMISR_0 Register (Offset = 3Ch) [reset = 0h]

CORE1_CURMISR_0 is shown in Figure 28-159 and described in Table 28-149.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-159. CORE1_CURMISR_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR0																															
R-0h																															

Table 28-149. CORE1_CURMISR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR0	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.17 CORE1_CURMISR_1 Register (Offset = 40h) [reset = 0h]

CORE1_CURMISR_1 is shown in Figure 28-160 and described in Table 28-150.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-160. CORE1_CURMISR_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR1																															
R-0h																															

Table 28-150. CORE1_CURMISR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR1	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.18 CORE1_CURMISR_2 Register (Offset = 44h) [reset = 0h]

CORE1_CURMISR_2 is shown in Figure 28-161 and described in Table 28-151.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-161. CORE1_CURMISR_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR2																															
R-0h																															

Table 28-151. CORE1_CURMISR_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR2	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.19 CORE1_CURMISR_3 Register (Offset = 48h) [reset = 0h]

CORE1_CURMISR_3 is shown in Figure 28-162 and described in Table 28-152.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-162. CORE1_CURMISR_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR3																															
R-0h																															

Table 28-152. CORE1_CURMISR_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR3	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.20 CORE1_CURMISR_4 Register (Offset = 4Ch) [reset = 0h]

CORE1_CURMISR_4 is shown in Figure 28-163 and described in Table 28-153.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-163. CORE1_CURMISR_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR4																															
R-0h																															

Table 28-153. CORE1_CURMISR_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR4	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.21 CORE1_CURMISR_5 Register (Offset = 50h) [reset = 0h]

CORE1_CURMISR_5 is shown in Figure 28-164 and described in Table 28-154.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-164. CORE1_CURMISR_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR5																															
R-0h																															

Table 28-154. CORE1_CURMISR_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR5	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.22 CORE1_CURMISR_6 Register (Offset = 54h) [reset = 0h]

CORE1_CURMISR_6 is shown in Figure 28-165 and described in Table 28-155.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-165. CORE1_CURMISR_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR6																															
R-0h																															

Table 28-155. CORE1_CURMISR_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR6	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.23 CORE1_CURMISR_7 Register (Offset = 58h) [reset = 0h]

CORE1_CURMISR_7 is shown in Figure 28-166 and described in Table 28-156.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-166. CORE1_CURMISR_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR7																															
R-0h																															

Table 28-156. CORE1_CURMISR_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR7	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.24 CORE1_CURMISR_8 Register (Offset = 5Ch) [reset = 0h]

CORE1_CURMISR_8 is shown in Figure 28-167 and described in Table 28-157.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-167. CORE1_CURMISR_8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR8																															
R-0h																															

Table 28-157. CORE1_CURMISR_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR8	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.25 CORE1_CURMISR_9 Register (Offset = 60h) [reset = 0h]

CORE1_CURMISR_9 is shown in Figure 28-168 and described in Table 28-158.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-168. CORE1_CURMISR_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR9																															
R-0h																															

Table 28-158. CORE1_CURMISR_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR9	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.26 CORE1_CURMISR_10 Register (Offset = 64h) [reset = 0h]

CORE1_CURMISR_10 is shown in Figure 28-169 and described in Table 28-159.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-169. CORE1_CURMISR_10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR10																															
R-0h																															

Table 28-159. CORE1_CURMISR_10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR10	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.27 CORE1_CURMISR_11 Register (Offset = 68h) [reset = 0h]

CORE1_CURMISR_11 is shown in Figure 28-170 and described in Table 28-160.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-170. CORE1_CURMISR_11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR11																															
R-0h																															

Table 28-160. CORE1_CURMISR_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR11	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.28 CORE1_CURMISR_12 Register (Offset = 6Ch) [reset = 0h]

CORE1_CURMISR_12 is shown in Figure 28-171 and described in Table 28-161.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-171. CORE1_CURMISR_12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR12																															
R-0h																															

Table 28-161. CORE1_CURMISR_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR12	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.29 CORE1_CURMISR_13 Register (Offset = 70h) [reset = 0h]

CORE1_CURMISR_13 is shown in Figure 28-172 and described in Table 28-162.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-172. CORE1_CURMISR_13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR13																															
R-0h																															

Table 28-162. CORE1_CURMISR_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR13	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.30 CORE1_CURMISR_14 Register (Offset = 74h) [reset = 0h]

CORE1_CURMISR_14 is shown in Figure 28-173 and described in Table 28-163.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-173. CORE1_CURMISR_14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR14																															
R-0h																															

Table 28-163. CORE1_CURMISR_14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR14	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.31 CORE1_CURMISR_15 Register (Offset = 78h) [reset = 0h]

CORE1_CURMISR_15 is shown in Figure 28-174 and described in Table 28-164.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-174. CORE1_CURMISR_15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR15																															
R-0h																															

Table 28-164. CORE1_CURMISR_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR15	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.32 CORE1_CURMISR_16 Register (Offset = 7Ch) [reset = 0h]

CORE1_CURMISR_16 is shown in Figure 28-175 and described in Table 28-165.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-175. CORE1_CURMISR_16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR16																															
R-0h																															

Table 28-165. CORE1_CURMISR_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR16	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.33 CORE1_CURMISR_17 Register (Offset = 80h) [reset = 0h]

CORE1_CURMISR_17 is shown in Figure 28-176 and described in Table 28-166.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-176. CORE1_CURMISR_17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR17																															
R-0h																															

Table 28-166. CORE1_CURMISR_17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR17	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.34 CORE1_CURMISR_18 Register (Offset = 84h) [reset = 0h]

CORE1_CURMISR_18 is shown in Figure 28-177 and described in Table 28-167.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-177. CORE1_CURMISR_18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR18																															
R-0h																															

Table 28-167. CORE1_CURMISR_18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR18	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.35 CORE1_CURMISR_19 Register (Offset = 88h) [reset = 0h]

CORE1_CURMISR_19 is shown in Figure 28-178 and described in Table 28-168.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-178. CORE1_CURMISR_19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR19																															
R-0h																															

Table 28-168. CORE1_CURMISR_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR19	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.36 CORE1_CURMISR_20 Register (Offset = 8Ch) [reset = 0h]

CORE1_CURMISR_20 is shown in Figure 28-179 and described in Table 28-169.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-179. CORE1_CURMISR_20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR20																															
R-0h																															

Table 28-169. CORE1_CURMISR_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR20	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.37 CORE1_CURMISR_21 Register (Offset = 90h) [reset = 0h]

CORE1_CURMISR_21 is shown in Figure 28-180 and described in Table 28-170.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-180. CORE1_CURMISR_21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR21																															
R-0h																															

Table 28-170. CORE1_CURMISR_21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR21	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.38 CORE1_CURMISR_22 Register (Offset = 94h) [reset = 0h]

CORE1_CURMISR_22 is shown in Figure 28-181 and described in Table 28-171.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-181. CORE1_CURMISR_22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR22																															
R-0h																															

Table 28-171. CORE1_CURMISR_22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR22	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.39 CORE1_CURMISR_23 Register (Offset = 98h) [reset = 0h]

CORE1_CURMISR_23 is shown in Figure 28-182 and described in Table 28-172.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-182. CORE1_CURMISR_23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR23																															
R-0h																															

Table 28-172. CORE1_CURMISR_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR23	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.40 CORE1_CURMISR_24 Register (Offset = 9Ch) [reset = 0h]

CORE1_CURMISR_24 is shown in Figure 28-183 and described in Table 28-173.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-183. CORE1_CURMISR_24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR24																															
R-0h																															

Table 28-173. CORE1_CURMISR_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR24	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.41 CORE1_CURMISR_25 Register (Offset = A0h) [reset = 0h]

CORE1_CURMISR_25 is shown in Figure 28-184 and described in Table 28-174.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-184. CORE1_CURMISR_25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR25																															
R-0h																															

Table 28-174. CORE1_CURMISR_25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR25	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.42 CORE1_CURMISR_26 Register (Offset = A4h) [reset = 0h]

CORE1_CURMISR_26 is shown in Figure 28-185 and described in Table 28-175.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-185. CORE1_CURMISR_26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR26																															
R-0h																															

Table 28-175. CORE1_CURMISR_26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR26	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.43 CORE1_CURMISR_27 Register (Offset = A8h) [reset = 0h]

CORE1_CURMISR_27 is shown in Figure 28-186 and described in Table 28-176.

Return to Summary Table.

Holds the MISR signature for CORE1

Figure 28-186. CORE1_CURMISR_27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1MISR27																															
R-0h																															

Table 28-176. CORE1_CURMISR_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C1MISR27	R	0h	MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.44 CORE2_CURMISR_0 Register (Offset = ACh) [reset = 0h]

CORE2_CURMISR_0 is shown in Figure 28-187 and described in Table 28-177.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-187. CORE2_CURMISR_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR0																															
R-0h																															

Table 28-177. CORE2_CURMISR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR0	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.45 CORE2_CURMISR_1 Register (Offset = B0h) [reset = 0h]

CORE2_CURMISR_1 is shown in Figure 28-188 and described in Table 28-178.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-188. CORE2_CURMISR_1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR1																															
R-0h																															

Table 28-178. CORE2_CURMISR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR1	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.46 CORE2_CURMISR_2 Register (Offset = B4h) [reset = 0h]

CORE2_CURMISR_2 is shown in Figure 28-189 and described in Table 28-179.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-189. CORE2_CURMISR_2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR2																															
R-0h																															

Table 28-179. CORE2_CURMISR_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR2	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.47 CORE2_CURMISR_3 Register (Offset = B8h) [reset = 0h]

CORE2_CURMISR_3 is shown in Figure 28-190 and described in Table 28-180.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-190. CORE2_CURMISR_3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR3																															
R-0h																															

Table 28-180. CORE2_CURMISR_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR3	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.48 CORE2_CURMISR_4 Register (Offset = BCh) [reset = 0h]

CORE2_CURMISR_4 is shown in Figure 28-191 and described in Table 28-181.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-191. CORE2_CURMISR_4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR4																															
R-0h																															

Table 28-181. CORE2_CURMISR_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR4	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.49 CORE2_CURMISR_5 Register (Offset = C0h) [reset = 0h]

CORE2_CURMISR_5 is shown in Figure 28-192 and described in Table 28-182.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-192. CORE2_CURMISR_5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR5																															
R-0h																															

Table 28-182. CORE2_CURMISR_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR5	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.50 CORE2_CURMISR_6 Register (Offset = C4h) [reset = 0h]

CORE2_CURMISR_6 is shown in Figure 28-193 and described in Table 28-183.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-193. CORE2_CURMISR_6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR6																															
R-0h																															

Table 28-183. CORE2_CURMISR_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR6	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.51 CORE2_CURMISR_7 Register (Offset = C8h) [reset = 0h]

CORE2_CURMISR_7 is shown in Figure 28-194 and described in Table 28-184.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-194. CORE2_CURMISR_7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR7																															
R-0h																															

Table 28-184. CORE2_CURMISR_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR7	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.52 CORE2_CURMISR_8 Register (Offset = CCh) [reset = 0h]

CORE2_CURMISR_8 is shown in Figure 28-195 and described in Table 28-185.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-195. CORE2_CURMISR_8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR8																															
R-0h																															

Table 28-185. CORE2_CURMISR_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR8	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.53 CORE2_CURMISR_9 Register (Offset = D0h) [reset = 0h]

CORE2_CURMISR_9 is shown in Figure 28-196 and described in Table 28-186.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-196. CORE2_CURMISR_9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR9																															
R-0h																															

Table 28-186. CORE2_CURMISR_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR9	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.54 CORE2_CURMISR_10 Register (Offset = D4h) [reset = 0h]

CORE2_CURMISR_10 is shown in Figure 28-197 and described in Table 28-187.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-197. CORE2_CURMISR_10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR10																															
R-0h																															

Table 28-187. CORE2_CURMISR_10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR10	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.55 CORE2_CURMISR_11 Register (Offset = D8h) [reset = 0h]

CORE2_CURMISR_11 is shown in Figure 28-198 and described in Table 28-188.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-198. CORE2_CURMISR_11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR11																															
R-0h																															

Table 28-188. CORE2_CURMISR_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR11	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.56 CORE2_CURMISR_12 Register (Offset = DCh) [reset = 0h]

CORE2_CURMISR_12 is shown in Figure 28-199 and described in Table 28-189.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-199. CORE2_CURMISR_12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR12																															
R-0h																															

Table 28-189. CORE2_CURMISR_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR12	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.57 CORE2_CURMISR_13 Register (Offset = E0h) [reset = 0h]

CORE2_CURMISR_13 is shown in Figure 28-200 and described in Table 28-190.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-200. CORE2_CURMISR_13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR13																															
R-0h																															

Table 28-190. CORE2_CURMISR_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR13	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.58 CORE2_CURMISR_14 Register (Offset = E4h) [reset = 0h]

CORE2_CURMISR_14 is shown in Figure 28-201 and described in Table 28-191.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-201. CORE2_CURMISR_14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR14																															
R-0h																															

Table 28-191. CORE2_CURMISR_14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR14	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.59 CORE2_CURMISR_15 Register (Offset = E8h) [reset = 0h]

CORE2_CURMISR_15 is shown in Figure 28-202 and described in Table 28-192.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-202. CORE2_CURMISR_15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR15																															
R-0h																															

Table 28-192. CORE2_CURMISR_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR15	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.60 CORE2_CURMISR_16 Register (Offset = ECh) [reset = 0h]

CORE2_CURMISR_16 is shown in Figure 28-203 and described in Table 28-193.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-203. CORE2_CURMISR_16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR16																															
R-0h																															

Table 28-193. CORE2_CURMISR_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR16	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.61 CORE2_CURMISR_17 Register (Offset = F0h) [reset = 0h]

CORE2_CURMISR_17 is shown in Figure 28-204 and described in Table 28-194.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-204. CORE2_CURMISR_17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR17																															
R-0h																															

Table 28-194. CORE2_CURMISR_17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR17	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.62 CORE2_CURMISR_18 Register (Offset = F4h) [reset = 0h]

CORE2_CURMISR_18 is shown in Figure 28-205 and described in Table 28-195.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-205. CORE2_CURMISR_18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR18																															
R-0h																															

Table 28-195. CORE2_CURMISR_18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR18	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.63 CORE2_CURMISR_19 Register (Offset = F8h) [reset = 0h]

CORE2_CURMISR_19 is shown in Figure 28-206 and described in Table 28-196.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-206. CORE2_CURMISR_19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR19																															
R-0h																															

Table 28-196. CORE2_CURMISR_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR19	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.64 CORE2_CURMISR_20 Register (Offset = FCh) [reset = 0h]

CORE2_CURMISR_20 is shown in Figure 28-207 and described in Table 28-197.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-207. CORE2_CURMISR_20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR20																															
R-0h																															

Table 28-197. CORE2_CURMISR_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR20	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.65 CORE2_CURMISR_21 Register (Offset = 100h) [reset = 0h]

CORE2_CURMISR_21 is shown in Figure 28-208 and described in Table 28-198.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-208. CORE2_CURMISR_21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR21																															
R-0h																															

Table 28-198. CORE2_CURMISR_21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR21	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.66 CORE2_CURMISR_22 Register (Offset = 104h) [reset = 0h]

CORE2_CURMISR_22 is shown in Figure 28-209 and described in Table 28-199.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-209. CORE2_CURMISR_22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR22																															
R-0h																															

Table 28-199. CORE2_CURMISR_22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR22	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.67 CORE2_CURMISR_23 Register (Offset = 108h) [reset = 0h]

CORE2_CURMISR_23 is shown in Figure 28-210 and described in Table 28-200.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-210. CORE2_CURMISR_23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR23																															
R-0h																															

Table 28-200. CORE2_CURMISR_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR23	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.68 CORE2_CURMISR_24 Register (Offset = 10Ch) [reset = 0h]

CORE2_CURMISR_24 is shown in Figure 28-211 and described in Table 28-201.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-211. CORE2_CURMISR_24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR24																															
R-0h																															

Table 28-201. CORE2_CURMISR_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR24	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.69 CORE2_CURMISR_25 Register (Offset = 110h) [reset = 0h]

CORE2_CURMISR_25 is shown in Figure 28-212 and described in Table 28-202.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-212. CORE2_CURMISR_25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR25																															
R-0h																															

Table 28-202. CORE2_CURMISR_25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR25	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.70 CORE2_CURMISR_26 Register (Offset = 114h) [reset = 0h]

CORE2_CURMISR_26 is shown in Figure 28-213 and described in Table 28-203.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-213. CORE2_CURMISR_26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR26																															
R-0h																															

Table 28-203. CORE2_CURMISR_26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR26	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.5.7.71 CORE2_CURMISR_27 Register (Offset = 118h) [reset = 0h]

CORE2_CURMISR_27 is shown in Figure 28-214 and described in Table 28-204.

Return to Summary Table.

Holds the MISR signature for CORE2

Figure 28-214. CORE2_CURMISR_27 Register

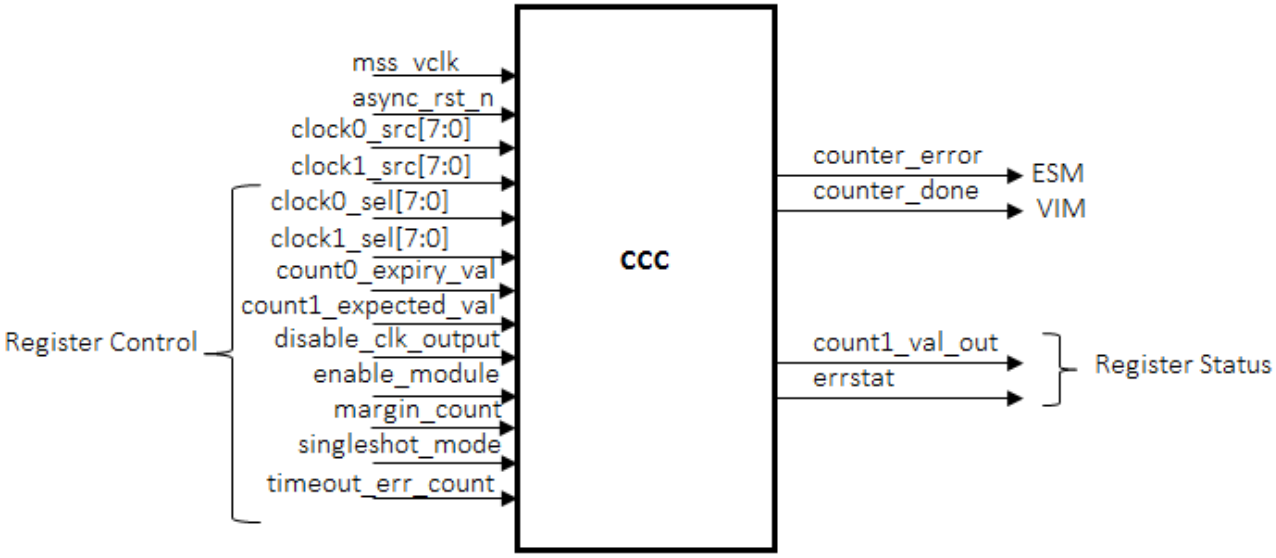
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2MISR27																															
R-0h																															

Table 28-204. CORE2_CURMISR_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	C2MISR27	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

28.6 Core Clock Comparator (CCC) Module

Figure 28-215. Core Clock Comparator Module



28.6.1 Description

CCC supports singleshot and continuous mode of an operation such as DCC. In continuous mode, the programmed values are reloaded after every successful comparison.

Module accepts 7 clock inputs for clock 0 and clock 1. One of these clock inputs is selected to counter 0 and counter 1. Counter 0 is a down counter, and is preloaded with a value before enabling the module. Counter 1 is an up counter which operates on clock 1.

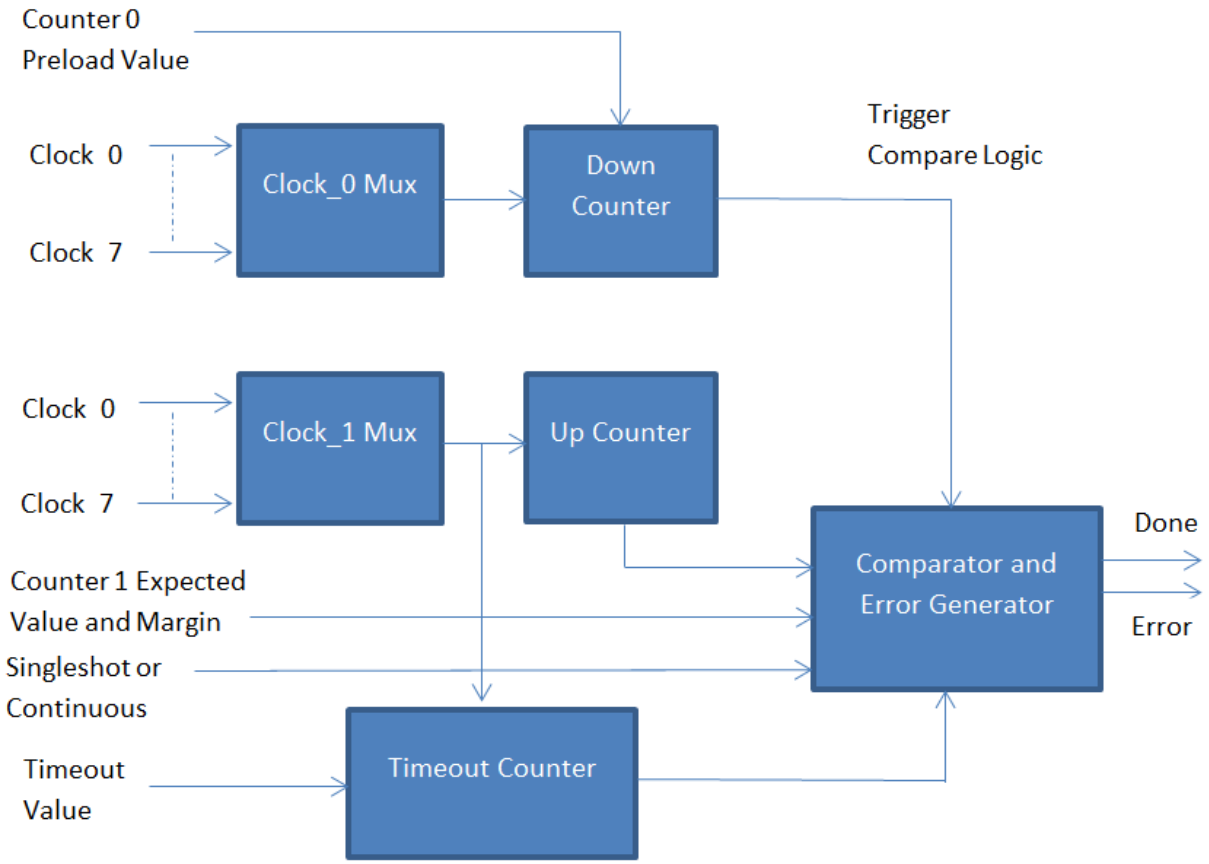
At the expiry of counter 0, value in the counter 1 is compared against the programmed expected value of the counter. After a successful comparison, a done signal is asserted in singleshot mode, whereas in continuous mode, counter 0 is reloaded for the next comparison. Margin value programming provides the tolerance for the comparison. An error signal is asserted when the counter 1 value differs from the expected value beyond the tolerance range.

When an error occurs, the module stops comparison in both singleshot and continuous mode.

There is a time-out counter functioning on clock 1. A time-out value must be loaded into the time-out counter before enabling the module. If the time-out counter expires before the expiry of counter 0, an error condition is indicated. In continuous mode, after the successful comparison, the time-out value is also reloaded along with counter 0.

28.6.2 Block Diagram

Figure 28-216. Block Diagram



28.6.3 Perform Clock Comparison

1. Select clock 0.
2. Select clock 1.
3. Load value for down counting in counter 0.
4. Load expected value of counter 1.
5. Load margin value for tolerance.
6. Set singleshot or continuous mode.
7. Load time-out value.
8. Enable module.

9. Wait for done or error indication.

28.6.4 Recommended Programmation

- Clock source 1 must be faster than clock source 0 for a successful comparison of clocks.
- The time-out value must always be loaded for successful comparison. The time-out value must be greater than the duration of comparison operation.
- CCC Registers: See the following registers in the MSS_GPCFG registers in the appropriate device's Control Register section.
  - CCCACFG0
  - CCCACFG1
  - CCCACFG2
  - CCCACFG3
  - CCCBCFG0
  - CCCBCFG1
  - CCCBCFG2
  - CCCBCFG3
  - CCCACNTVAL
  - CCCBCNTVAL
  - CCCABERRSTAT
  - CCCBWDEN (applicable for the 16xx/18xx device only)

Glossary

1

1MS— One milisecond

2

2D— Two Dimensional

3

3A— AF + AE + AWB

3D— Three Dimensional

3G— 3rd Generation of mobile communication systems

3GPP— 3rd Generation Partnership Project

8

8B— 8 bits

10

10B/8B— 10 bit to 8 bit conversion (i.e., 8B/10B decoding)

A

A/D— Analog to Digital Converter

ABB— Adaptive Body Bias

ABBLDO— Adaptive Body Bias Low Drop Out

ABS— Absolute Value

ACB— AC–bias frequency

ACBI— AC–bias line transitions per interrupt

ACE— ASIC Compiler Environment.

ACK— Acknowledge

ADC— Analog-to-Digital Converter/Conversion.

ADMA— Advanced DMA

ADPLL— All Digital Phase Locked Loop. A closed loop frequency control system whose function is based on the phase–sensitive detection of the phase difference between the input signal and the output signal of the controlled oscillator (CO).

AER— Advanced Error Reporting (PCIe)

AES— Advanced Encryption Standard

AET— Advanced Event Triggering - this capability can be used to debug complex problems as well as understand performance characteristics of user applications

AF— Auto Focus

AFE— Analog Front End

AFSX— Audio transmit frame synchronization

AHB— Advanced High-performance Bus

AHB-AP— Advanced High-performance Bus - Access Port

AIC— Analog Interface Chip.

ALE— Address Latch Enable

AM— Amplitude Modulation

AMBA— Advanced Micro-Controller Bus Architecture

AMMU— Attribute Memory Management Unit

ANSI— American National Standards Institute

AP— Address Protection

APB— Advanced Peripheral Bus

APB-AP— Advanced Peripheral Bus - Access Port

APE— Application Engine

API— Application Programming Interface

APLL— Analog Phase–Locked Loop

APWM— Asymmetrical Pulse Width Modulation (eCAP module related)

AR— Automatic Reload

ARGB— Alpha Red Green Blue

ARI— Alternative Routing-ID Interpretation (PCIe)

ARM— Advanced RISC Machine

ASCII— American Standard Code for Information Interchange

ASIC— Application–Specific Integrated Circuit

ASP— Application-Specific Peripheral

ASPM— Active State Power Management: Automatic PM in PCIe link layer FSM (PCIe)

ATA— Interface standard for the connection of storage devices (Advanced Technology Attachment)

ATAPI— ATA Packet Interface

ATB— AMBA Advanced Trace Bus

ATS— Address Translation Service (PCIe)

AVB— Audio Video Bridging

AVS— Adaptive Voltage Scaling

AWB— Auto White Balance

AWS— Audio Word Select

AXI— Adavced eXtensible Interface

B

B— Byte, 8 bits

BARn— Base Address Register (PCIe)

BB— Bus Busy

BCD— Binary-Coded Decimal.

BCH— Bose-Chaudhuri-Hocquenghem (CRC Code)

BCM— BIST Combiner Module

BDF— Bus / Device/ Function: (PCIe) function identifier

BE— Big Endian.

BFSW— Buffer Switch

BGA— Ball Grid Array

BGAP— Band Gap

BGT— Block Guard Time

BIOS— Basic Input/Output System. First software run after reset. Known also as ROM Code.

BIST— Built-In Self-Test

BIU— Bus Interface Unit

BL— Buffer Logic

BME— Bus Master Enable (PCI CFG bit)

BNC— British Naval Connector, Bayonet Nut Connector, or Bayonet Neill Concelman

BOF— Beginning of Frame

BOL— Beginning of Line

BPP— Bits Per Pixel

BS— Block Synchronization

BSP— Buffered Serial Port.

BTA— Bus Turn Around

BW— Band Width

BWM— Band Width Manager (functional entity in the TMS320C66x DSP CorePac)

BWS— Baseband Word Select

C

CABAC— Context-Adaptive Binary Arithmetic Coding

CAF— Combing Artifacts Calculation

CALC3— Transform and Quantization Calculation Engine

CAN— Controller Area Network

CAS— Column Address Strobe

CAVLC— Context-Adaptive Variable Length Coder

CB— Copy Back

CBC— Cipher Block Chaining

CBP— Coded Block Pattern

CBUF— Circular Buffer (Associated with OCMC)

CC— Channel Controller

CCC— Command Completion Coalescing feature

CDP— Coprocessor Data Operation

CDR— Clock Data Recovery

CE— Chip Enable

CEA-861-D— Video standard. It defines the video timing requirements, discovery structures, and data transfer structure.

CEC— Consumer Electronics Control

CFI— Common Flash Interface device

CH— Configuration Header. To use different settings than ROM Code defaults, i.e. clock frequencies, SDRAM settings, GPMC settings.

CID— Card Identification Number

CIF— Common Intermediate Format.

CIR— Consumer Infra Red

CLB— Command List Base Address

CLE— Command Latch Enable

CLK— Clock

CLO— Command List Override feature

CLUT— Color Look-Up Table

CM— Clock Management

CMOS— Complimentary Metal Oxide Semiconductor

CO— Controlled Oscillator

CODEC— Coder/Decoder or Compression/Decompression. A device that codes in one direction of transmission and decodes in another direction of transmission.

COS— Class Of Service

CP15— Arm® Cortex®-A15 system control coprocessor. This coprocessor controls and provides status information for the functions implemented in the main processor

CPFROM— Customer Programable eFuse ROM

CPPI— Communications Port Programming Interface

CPR— Color Phase Rotation

CPS— Cell Search Platform

CPSR— Current Program Status Register

CPU— Central Processing Unit

CRC— Cyclic Redundancy Check

CS— Chip-Select

CSL— Chip Support Library

CSWR— Closed switch retention

CTM— Counter-Timer Module

CTR— Hardware block for CCTrCh processing

CTRL— Control

CTS— Clear to Send

ConnID— Connection Identifier.An Initiator Module Identifier. A ConnID is transmitted in-band with the request and is used for security and error logging mechanism.

D

D2H— Device to Host transfer of a FIS

DAC— Digital to Analog Converter

DAP— Debug Access Port

DBI— Display Buffer Interface

DC— Direct Current

DCAN— CAN Controller Module

DCC— Duty Cycle Correction Circuit

DCD— Data Carrier Detect

DCO— Digitally Controlled Oscillator

DCT— Discrete Cosine Transform.

DDC— Display Data Channel

DDR— Double Data Rate

DE— Data Enable

DED— Double Error Detection

DEI— Deinterlacer

DES— Data Encryption Standard.

DES3DES— Data Encryption Standard and triple DES

DFF— Data Flip-Flop

DFT— Design For Test

DIN— Data In

DIR— Digital audio Interface Receiver

DISPC— Display Controller

DL— Data Length

DLB— Data Loopback.

DLL— Delay-Locked Loop

DLLP— Data Link Layer Packet: PCIe (compare PLP, TLP)

DMA— Direct Memory Access.

DMC— Digital Motor Control

DMEM— Data Memory

DMM— Dynamic Memory Management

DMT— Display Monitor Timing

DO— Data Out

DPC— Defect Pixel Correction

DPI— Display Parallel Interface

DPLL— Digital Phase-Locked Loop. Digital implementation of PLL.

DPS— Digital Power Switching

DRDY— Data Ready

DRM— Digital Rights Management

DS— Downstream: from the host (or a hub) to a device (or hub).

DSP— Digital Signal Processor.

DSR— Data Set Ready

DSS— Display Sub-System. Also DISS

DTBC— Data Buffer Controller

DTC— Debug and Trace Controller

DTCM— Data Tightly Coupled Memory

DTR— Data Transmit Ready

DVFS— Dynamic Voltage and Frequency Scaling

DVI— Digital Video Interface

DWORD— Double WORD (32-bit sized portion of data)

E

EAV— End of Active Video

ECAM— Enhanced Configuration Access Mechanism (PCIe)

ECC— Error Checking and Correction. Also Error Correction Code.

ECD3— Entropy Coder/Decoder

ECRC— End-to-end Cyclic Redundancy Check (optional, compare LCRC)

ED— Endpoint Descriptor

EDC— Error Detection Code

EDI— Edge Directed Interpolation

EDMA— Enhanced DMA

EDMA_TPCC— Enhanced DMA Third Party Channel Controller

EDMA_TPTC— Enhanced DMA Third Party Transfer Controller

EEE— Energy Efficient Ethernet

EEPROM— Electrically Erasable Programmable Read Only Memory

EFUSE— Electrical Fuse. A one-time programmable memory location usually set at the factory

EMC— External Memory Controller

EMI— Electromagneric interference

EMIF— External Memory Interface

EOB— End of Block

EOF— End of Frame

EOL— End of Line

EOP— End of Packet

EOT— End of Transfer

EP— Entry Point. Also End Point in USB and PCIe controller context.

EPM— Emulation Pin Manager

ES— Erase Status

ESC— Escape

ESR— Event Set Register

ETB— Embdedded Trace Bus

ETM— Embedded Trace Macrocell

EVM— Evaluation Module

F

FB— Received FISes base address

FBB— Forward Body-Bias

FC— Frame Counter

FCLK— Functional Clock

FCS— Frame Check Sequence

FD— Face Detect

FE— Framing Error

FF— Flip-Flop

FIFO— First In First Out

FIQ— Fast Interrupt Request. See ISR.

FIR— Fast Infrared

FLR— Function Level Reset (PCIe)

FM— Frequency Modulation

FMD— Film Mode Detection

FPGA— Field Programmable Gate Array.

FPKA— Fast-Public Key Accelerator

FROM— eFuse ROM

FS— Full-Speed

FSM— Finite State Machine.

FTS— Fast Training Sequence (PCIe)

FW— Firewall

G

GDP— Generic Dot Product

GFX— Graphics

GHB— Global History Buffer

GIC— Secure interrupt controller

GMII— Gigabit Media Independent Interface

GP— General-purpose

GPIO— General Purpose Input/Output

GPMC— General Purpose Memory Controller

GPU— Graphics-processing unit

GT/s— Gigatransfers per second

GZ— Ground Zero

H

H.263— Video Codec Standard

H.264— Video Codec Standard

H/W— Hardware

H2D— Host to Device transfer of a FIS

HAL— Hardware Abstraction Layer

HBA— Host bus adapter

HBP— Horizontal Back Porch

HC— Host Controller

HCI— Host Controller Interface

HD— High Definition

HDCP— High-bandwidth Digital Content Protection

HDD— Hard-disk drive

HDMI— High-Definition Multimedia Interface

HDTV— High-Definition Television

HFP— Horizontal Front Porch

HLOS— High-Level Operating System

HMI— Human Machine Interface

HNP— Host Negotiation Protocol (OTG feature)

HPF— High-Pass Filter

HS— High-Speed

HSDPA— High Speed Downlink Packet Access

HSSCLL— High-Speed Serial Control Channel

HSUPA— High Speed Uplink Packet Access

HSW— Horizontal Synchronization Pulse Width

HSYNC— Horizontal Synchronization.

HW— Hardware

HWA— Hardware Accelerators.

HWOBS— Hardware Observability

I

I/F— Interface

I/O— Input/Output

I2C— Inter-Integrated Circuit

I2S— Inter-IC Sound

IA— Identifier Address

IATU— Internal Address Translation Unit (PCIe controller)

IBR— An ATU Inbound Region (PCIe)

ICE— In-Circuit Emulation

ICEPICK— Generic TAP for emulation control

ICLK— Interface Clock

ICONT— Imaging Controller

ICR— Intersystem Communication Registers

ID— Identification

IDCT— Inverse Discrete Cosine Transform. See DCT.

IDE— Integrated Development Environment. A programming environment integrated into an application.

IDO— ID-based Ordering (PCIe)

IIS— Inter–IC Sound

ILF3— Improved Loop Filter engine

IM— Initiator Module. A module is an initiator whenever it is able to initiate read and write requests to the chip interconnect (typically: processors, DMA, etc.).

IME3— Improved Motion Estimation engine

INT— Interrupt

INTC— Interrupt Controller

IP— Intellectual Property

IPC— Interprocessor Communication. Also referred to as "mailbox" on occasion

IPE3— Intra Prediction Estimation engine

IPU— Image-processing unit

IQ— Inverse Quantization

IR— Incremental Redundancy Buffer

IRQ— Interrupt Request.

ISA— Instruction Set Architecture

ISE— I/O Space Enable: (PCI CFG bit)

ISO— Isochronous.

ISP— Image Signal Processor

ISR— Interrupt Service Routine.

IST— Interrupt Service Thread.

ITCCHEN— Intermediate transfer completion chaining enable.

ITCINTEN— Intermediate transfer completion interrupt enable.

ITCM— Instruction Tightly Coupled Memory

IVA— Image and Video Accelerator

IVA-HD— High Definition Image and Video Accelerator

IrDA— Infrared Data Association.

J

JEDEC— Joint Electronic Devices Engineering Council

JPEG— Joint Photographics Experts Group

JTAG— Joint Test Action Group.

K

KB— Kilobyte, 1024 Bytes

Kbps— Kilobits per second

L

L1— Level 1 cache/memory

L2— Level 2 cache/memory

L3— First level of interconnect

L4— Second level of interconnect

LA— Logical Address

LAN— Local Area Network

LBA— Logical Block Addressing

LC— Logical Channel

LCD— Liquid Crystal Display.

LCRC— Link Cyclic Redundancy Check (mandatory, compare ECRC)

LCh— Logical DMA Channel. Also LCH

LD— Lens Distortion

LDO— Low Dropout

LE— Little Endian.

LEC— Line End Code

LED— Light Emitting Diode.

LF— Loop Filter

LFPS— Low Frequency Periodic Signaling defined in USB 3.0 SuperSpeed standard

LFSR— Linear-Feedback Shift Register

LH— Local Host

LINK— Link Layer Device

LISA— Local Interconnect and Synchronization Agent

LLC— Link Layer Control

LLP— Low-Level Protocol

LP— Low-Power, operation mode for PHY

LPAE— Large Physical Address Extensions

LPM— Low Power Mode (also known as LPMODE)

LPP— Lines Per Panel

LRU— Least Recently Used

LS— Low-Speed

LSB— Least Significant Bit

LSE— IVA Load and Store Engine

LSM— LISA Section Manager

LSR— Linear Shift Register

LSW— Least Significant Word

LTR— Latency Tolerance Reporting (PCIe)

LTSSM— Link Training and Status State Machine: In PCIe - PM FSM of PCIe physical layer

LUT— Look-up Table

LVCMOS— Low Voltage Complementary Metal Oxide Semiconductor

LVDS— Low-Voltage Differential Signaling

M

M2— Micro Memory

MAC— Message Authentication Code

MAP— Maximum a posteriori algorithm

MB— Megabyte, 1024 KB

MBAFF— MB-Level Adaptive Frame/Field

MBMS— Multimedia Broadcast Multicast Service

MC3— Motion Compensation Engine

McSPI— Multichannel Serial Peripheral Interface

MCU— Microcontroller Unit. Refers to the MPU.

MDIO— Management Data Input/Output

MDMA— Modem DMA. Also Master DMA port of the DSPSS TMS320C66x CorePac.

MDT— Motion Detection

ME— Motion Estimation

MEMIF— Memory Interface

MIF— Memory InterFace

MII— Media Independent Interface

MIPI— Mobile Industry Processor Interface

MIR— Medium Infrared

MJPEG— Motion JPEG

MMR— Memory Mapped Register

MMU— Memory Management Unit.

MP3— MPEG Layer 3.

MPEG— Motion Pictures Expert Group.

MPEG1— The first MPEG compression scheme specification.

MPS— Maximum Packet Size

MPU— Microprocessor Unit.

MRRS— Maximum Read Request Size (PCIe)

MS— Memory Stick

MSB— Most Significant Bit

MSGIF— Message Interface

MSS— Master Subsystem

MUX— Multiplex/Multiplexer

MV— Motion Value

Mb— Megabit

Mbps— Mega bits per second

Modem— Modulator Demodulator

MuxMode— 3 bits field of the PIN/PAD Control register field which enables to change the mode. Mode programming is assumed by software and selects a function on the device external interface.

N

N/A— Not Applicable. Also Not Available.

NAC— Network Access Control

NAND— NAND Flash memory.

NIU— Network Interface Unit

NMI— Nonmaskable Interrupt. An interrupt that can be neither masked nor disabled.

NOP— No OPeration (DSP/CPU instruction)

NOR— A type of flash memory

NRZ— Non-Return-to-Zero

NTSC— National Television System Committee. Television broadcast system.

NVIC— Nested Vectored Interrupt Controller

O

OCM— On-chip Memory

OCMC— On-chip Memory Controller

OCP— Open-Core Protocol

OE— Output Enable

OOB— Out of Band signaling

OPP— Operating Performance Point

OS— Operating System

P

PA— Program Address

PATA— Parallel-ATA

PBIAS— PMOS Bias transistor to provide the bias voltage to extended drain IOs

PC— Program Counter

PCI— Peripheral Component Interconnect.

PCLK— Pixel Clock

PCM— Pulse Code Modulation.

PCS— Physical Coding Sublayer (component related to PCIePHY)

PD— Program Data

PDA— Personal Digital Assistant

PDC— Power-down Controller

PE— Parity Error

PFPW— Prefetch and Prewrite posting engine

PHY— Physical Layer Device

PI— Pixel Interpolation

PID— Protocol Identifier. The PID register is used in Windows CE mode only.

PIO— A programmed input/output transfer supported by IDE devices

PIPE3— Physical Interface for PCI Express rev.3.0

PLD— Programmable Logic Devices

PLL— Phase–Locked Loop.

PM— Programming Model

PMA— Physical Media Attachment sublayer

PMEM— Program Memory

PMFW— Power Management FrameWork

PMIC— Power management Integrated Circuit

PMP— Power Management Port

PMU— Performance Monitoring Unit

POR— Power–On Reset

PPA— Primary Protected Application.

PPC— Palm–size PC

PPI— Physical Layer Protocol Interface

PPL— Pixels per Line

PPM— Parts per million

PRCM— Power, Reset and Clock Management

PRD— Physical Region Descriptor

PRM— Power and Reset manager

PRT— PRD Table

PS— Packet Start

PSC— Prescaler Counter

PSS— Program Suspend Status

PT— Packet Type

PTI— Parallel Trace Interface

PTV— Prescale Clock Timer Value. Sets the value of the divisor used in scaling the clock.

PWL— Pulse Width Light (modulator). A 4096-bit randomsequence generator that provides control of the LCD backlighting and keypad.

PWM— Pulse Width Modulation

PaRAM— Programmable RAM that stores 32-byte channel transfer definition that EDMA channels, QDMA channels, and linking uses.

Q

QDMA— Quick DMA

QIQ— Quantization and Inverse Quantization

QOS— Quality of Service

QP— Quantization Parameter

QSPI— Quad Serial Peripheral Interface

R

R/W— Read/Write. Also RW.

R5— Release 5 of 3GPP specifications on IMS and HSDPA standards

R6— Release 6 of 3GPP specifications on Wireless LAN networks, HSUPA, MBMS and enhancements to IMS standards

RAM— Random Access Memory.Fast-access but volatile memory type.

RC— PCIe Root Complex

RDR— Receive Data Register

RE— Read Enable

REQ— Request

RF— Radio Frequency

RFF— Retention Flip-Flop

RFU— Reserved for Future Use

RGB— Red Green Blue

RGBA— Red Green Blue Alpha

RGMII— Reduced Gigabit Media Independent Interface

RI— Ring Indicator

RISC— Reduced Instruction Set Computing. A CPU whose instruction set and related decode mechanism are much simpler than those of microprogrammed complex instruction set computers. The result is a higher instruction throughput and a faster real–time interrupt service response from a smaller, cost-effective chip.

RM— Reed-Muller code

RMII— Reduced Media Independent Interface

RNG— Random Number Generator

RO— Read Only

ROM— Read Only Memory. A semiconductor storage element containing permanent data that cannot be changed.

RSS— Reset System Simulator

RST— Reset

RT— Real–Time

RTA— Retention Till Access

RTL— Register Transfert Level

RTOS— Real–Time Operating System

RTS— Request to Send

RV— RealVideo (codec)

RVLC— Reversible Variable Length Coder

RX— Receive/Receiver

RXD— Receive Data

RXN— PHY differential receiver (de-serializer) negative line

RXP— PHY differential receiver (de-serializer) positive line

S

S/PDIF— Sony/Philips Digital Interface

S/W— Software

SAF— Standby Associated Function

SAM— Signal Amplitude Modulation

SAR— Save and Restore. Hardware context saving for power saving.

SAV— Start of Active Video

SB— Silicon Backplane (Trade Mark)

SBC— Stream Buffer Controller

SBH— Synchronization Box Handler

SCL— Serial Clock. Programmable serial clock used in the I2C interface. Also SCLK.

SCM— System Control Module

SCP— Serial Configuration Port

SCTM— System Counter Timer Module

SCU— Snoop Control Unit

SD— Also Standard Definition (television system).

SDA— Serial Data. Serial data bus in the I2C interface.

SDMA— The Slave DMA port of the DSPSS TMS320C66x CorePac

SDP— Software Development Platform

SDR— Single Data Rate

SDRAM— Synchronous Dynamic Random Access Memory

SDRC— SDRAM Controller.

SDTV— Standard Definition Television

SE— Secure Environment.Execution environment inside a device, which is protected against tampering

SEC— Single Error Correction

SECDED— Single-Error Correction Dual-Error Detection

SFI— Serial Flash Interface

SGX— Accronym for Grafics Accelerator

SIF— Source Input Format

SIMD— Single Instruction-Stream, Multiple Data-Stream

SIR— Slow Infrared

SL2— Shared Level 2 (memory/interface)

SL2IF— Shared L2 Interface

SLC— Single Level Cell devices

SLM— Static Leakage Management

SMC— Secure Monitor Call

SMP— Symmetric Multiprocessor Platform

SMPS— Switch Mode Power Supply

SMSET— Software Message and System Event Trace module

SOC— System-On-a-Chip

SOF— Start Of Frame

SP— Serial Port or Small Page

SPC— Serial Port Control

SPI— Serial Peripheral Interface. A signaling protocol for exchanging serial data.

SR— SmartReflex

SR3-APG— SmartReflex3 Automatic Power Gating

SRAM— Static Random Access Memory

SRC— Sample Rate Conversion

SRMD— Single Request, Multiple Data

SRP— Session Request Protocol (OTG feature)

SS— Subsystem

SSC— Spread Spectrum Clocking

SSD— Solid State Drive

ST— Start Timer

STC— Store from Coprocessor (to memory) or System Time Clock, which is the master clock in an MPEG2 encoder or decoder system.

STM— Synchronous Transfer Mode or Store Multiple.

STN— Super-Twist Nematic. A technique for improving LCD display screens by twisting light rays.

SVC— Supervisor Call

SW— Software

SWI— Software Interrupt

SXGA— Super eXtended Graphics Array

SYNCDIM— Transfer synchronization dimension.

SYSCTRL— IVA System Control module

T

TA— Target Agent

TAP— Test Access Port

TC— Traffic Controller. Allows asynchronous operation among the external memory interface, the MPU, and the DSP. Also, in PCI Express context this means Traffic Class, mapped to a virtual channel

TCCHEN— Transfer complete chaining enable.

TCINTEN— Transfer complete interrupt enable.

TCK— Test Clock

TCM— Tightly Coupled Memory

TD— Transfer Descriptor, in PCI Express this means TLP Digest (see also ECRC)

TDI— Test Data Input

TDM— Time Division Multiplex/Multiplexing

TDO— Test Data Output

TERC4— TMDS Error Reduction Coding

TFT— Thin Film Transistor. A type of LCD flat panel display screen in which each pixel is controlled by one to four transistors.

TI— Texas Instruments

TILER— Tiling Isometric Lightweight Engine for Rotation

TLB— Translation Lookaside Buffer. A cache that contains entries for virtual-to-physical address translation and access permission checking.

TLP— Transaction Layer Packet: PCIe (compare PLP, DLLP)

TM— Target Module. A target module cannot generate read/write requests to the chip interconnects, but respond to these requests. However it may generate interrupts or DMA request to the system (typically: peripherals, memory controllers).

TMDS— Transition Minimized Differential Signaling. A technology for transmitting high-speed serial data and is used by the DVI and HDMI video interfaces.

TMS— Test Mode Select

TOC— Table of Contents

TP— Tiny Page

TPH— TLP Processing Hints (PCIe)

TPIU— Trace Port Interface Unit

TR— Tranfer Request

TRM— Technical Reference Manual

TRST— Test Reset

TS— Transmission Start

TSHUT— Temperature Shutdown.

TTL— Transistor Transistor Logic

TWL— Table Walking Logic

TX— Transmit//Transmitter

TXD— Transmit Data

TXN— PHY differential transmitter (serializer) negative line

TXP— PHY differential transmitter (serializer) positive line

U

UART— Universal Asynchronous Receiver/Transmitter. Another name for the asynchronous serial port.

USB— Universal Serial Bus.

USSE— Universal Scalable Shader Engine

V

VA— Virtual Address

VBP— Vertical Back Porch

VBUF— Virtual Frame Buffer

VC— Virtual Channel

VC-1— Video Codec Standard

VCA— Via Channel Array technology

VCO— Voltage Controlled Oscillator

VDMA— Video Direct Memory Access module

VESA— Video Electronics Standards Association

VFP— Vertical Front Porch

VGA— Video Graphics Array. An industry standard for video cards.

VID— VLAN Identifier

VIP— Video Input Port

VLAN— Virtual Local Area Network

VLC— Variable Length Decoder

VLD— Variable Length Coder

VLIW— Very Long Instruction Word

VMODE— Bi-level voltage control interface

VPE— Video Processing Engine

VR— Voice Recognition

VS— Vertical Synchronization

VSW— Vertical Synchronization Pulse Width

VSYNC— Vertical Synchronization. A bidirectional vertical timing signal occurring once per frame with a pulse-width defined as an integral number of lines (half-lines for interlaced mode). Also VS.

W

WB— Write Buffer

WC— Word Count

WD— Watchdog. A timer that requires the user program or OS periodically write to the count register before the counter underflows and triggers a reset.

WDT— Watchdog Timer. See WD.

WE— Write Enable

WFI— Wait For Interrupt

WIR— Wait In Reset

WNP— Write Non-Posted

WP— Write Protect

WT— Write Through

WTBU— Wireless Terminal Business Unit

Word16— 16 bits word

X

X-LOADER— A user-defined pre-operating system bootstrap code that resides at the beginning of the external flash.

XIP— eXecution In Place

XMC— Extended Memory Controller (a functional entity in the TMS320C66X DSP CorePac)

Y

YUV— Luminance-Bandwidth-Chrominance

e

eCAP— Enhanced Capture Module

x

xHCI— Extensible Host Controller Interface

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from September 1, 2019 to June 30, 2020 (from D Revision (September 2019) to E Revision)	Page
• Added 64xx information.	247
• Added DSS_HW_ACC_MC_PING to Master Subsystem, Cortex-R4F Memory Map table.	253
• Added DSS_HW_ACC_MC_PONG to Master Subsystem, Cortex-R4F Memory Map table.	253
• Added MSS_RTID to Master Subsystem, Cortex-R4F Memory Map table.	255
• Added MSS_RTIC to Master Subsystem, Cortex-R4F Memory Map table.	255
• Added DSS_HW_ACC_MC_PING to DSP C674x Memory Map table.	258
• Added DSS_HW_ACC_MC_PONG to DSP C674x Memory Map table.	258
• Updated SYSRSTCAUSE Register Field Descriptions.	310
• Updated RSTCAUSE Register Field Descriptions.	369
• Added SPARE0_3 register.	488
• Added SPARE4_9 register.	488
• Updated SYSRSTCAUSE Register Field Descriptions.	497
• Updated CLKSRCSEL0 Register.	539
• Updated CLKGATE Register.	541
• Added RTICURRCLKDIV Register.	544
• Updated RSTCAUSE Register Field Descriptions.	569
• Updated CLKINUSE Register .	572
• Updated ECCCAPTMSSBSS Register Field Description.	574
• Added GPCFG6 Register.	584
• Added SPARE0_3 register.	787
• Added SPARE4_9 register.	787
• Updated SYSRSTCAUSE Register Field Descriptions.	796
• Updated RSTCAUSE Register Field Descriptions.	878
• Added GPCFG6 Register.	895
• Added SPARE0_3 register.	1097
• Added SPARE4_9 register.	1097
• Updated SYSRSTCAUSE Register Field Descriptions.	1106
• Updated CLKSRCSEL0 Register.	1148
• Updated CLKGATE Register.	1150
• Added RTICURRCLKDIV Register.	1153
• Updated RSTCAUSE Register Field Descriptions.	1178
• Updated CLKINUSE Register .	1181
• Updated ECCCAPTMSSBSS Register Field Description.	1183
• Added GPCFG6 Register.	1193
• Updated Transmit Subsystem section.	1398
• Updated Receive Subsystem section.	1399
• Removed EDMA_TPTC Configuration section.	1603
• Updated Reset values and Field Descriptions of EDMA_TPCC_CCCFG Register.	1621
• Updated STAT bit description in EDMA_TPTC_ERRDET Register Field Descriptions table.	1820
• Updated ADC Buffer Single-Chirp and Multi-Chirp Mode Programming Sequence table.	1847
• Updated ADC Buffer Continuous Mode Programming Sequence table.	1848
• Updated MSS_GIO Registers.	2160
• Updated TBCTL_CLKDIV bit description in TBCTL_TBSTS Register Field Descriptions table.	2384
• Updated TBCTL_HSPCLKDIV bit description in TBCTL_TBSTS Register Field Descriptions table.	2384
• Updated MCAN Block Diagram.	2601
• Updated MCAN Functional Description section.	2601
• Updated SPI_DC Register Field Descriptions table.	3177
• Updated SWnRST bit description in the SCI Global Control Register 1 (SCIGCR1).	3253

- Updated CRC Registers. 3355

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265