Technical Reference Manual

xWRLx432 Technical Reference Manual

TEXAS INSTRUMENTS

ADVANCE INFORMATION

Table of Contents

ADVANCE INFORMATION

ADVANCE INFORMATION

ADVANCE INFORMATION

ADVANCE INFORMATION

Trademarks

Spansion® is a registered trademark of Spansion LLC.
Macronix® is a registered trademark of Macronix International Co., Ltd.
All trademarks are the property of their respective owners.

ADVANCE INFORMATION

1 Introduction

This document contains information applicable to the following devices

- IWRL6432
- IWRL1432
- AWRL6432
- AWRL1432
- IWRL6432AOP

Compatability Note: Information throughout this document regarding the IWRL6432 devices is also applicable to the IWRL6432AOP devices

The xWRLx432 device is a family of single chip and ultra-low power 60 and 77GHz radar devices with 3 RXs and 2 TXs. This chapter introduces the features, subsystems, and architecture of xWRLx432 Systems on Chip (SoCs).

1.1 Device Overview

The xWRLx432 is targeted for building automation, displacement transmitters, appliances and laptops in the industrial space, and parking assist, in-cabin applications and hands-free access in the automotive space. The SoC has been designed as a low-power, high performance, and highly integrated device, with significant enhancements enabling ultra-low power use-cases.Some of the main distinguished characteristics of the device are:

- Single-chip radar transceiver with integrated LO, with 3 RX and 2 TX
- Support for 57-64 GHz(xWRL6432) and 77-81GHz (xWRL1432)
- Includes integrated ARM Cortex M4F @160MHz, ARM Cortex M3F @80MHz and Radar Hardware Accelerator (HWA) @80MHz for radar processing
- Up to 1 MB of on-chip RAM memory split across APPSS and the shared memory bank
- RF capabilities
  - Closed-loop frequency synthesis supporting ramp rates up to 500MHz/us
  - IF bandwidth up to 5MHz, ADC sampling rate up to 12.5Msps
  - TX binary phase modulation (BPM)
- ROM boot loader to configure the front-end and abstracted by mmWaveLink API layer which supports QSPI-Flash, UART and SPI based image download.
- Enhanced power capabilities and ultra-low power operation
  - Top level control of all subsystem power domains
  - Deep sleep mode for ultra-low power consumption
  - Low active power consumption
  - Integrated ROM boot loader completely eliminates the secondary boot loader and saves power
  - Light wight and low latency mmWaveLink APIs to support low power
- Debug capabilities.

The device provides a rich set of peripherals, such as:

- General connectivity peripherals, including:
  - One Inter-Integrated Circuit (I2C) interface
  - Two Controller/Peripheral Serial Peripheral Interfaces (SPI)
  - Two configurable Universal Asynchronous Receiver/Transmitter (UART) interfaces
  - One General-Purpose Input/Output (GPIO) module
  - One 2-external channel 8-bit Analog to Digital Convertor (GPADC), with an ADC Buffer size of 2 KB
- High-speed interfaces, including:
  - One RDIF for raw data capture of upto 100Mbps [4 Data, 2 Clocks]
- Control and Communication interfaces, including:
  - Two Controller Area Network (CAN-FD) interfaces with full flexible data rate support
  - One Enhanced Pulse Width Modulation (EPWM) modules

- External Memory (EMIF) Interface
  - One Quad-Serial Peripheral Interface (QSPI) at up to 80 MHz
- Timers and Watchdog Module
  - One Real Time Interrupts (RTI) modules
  - One Watchdog modules (Same RTI IP but in Watchdog configuration)
- Interprocesser Communication (IPC) interface integrated with DFP

The device includes different modules for functional safety requirements support:

- Logic BIST mechanism for all the CPU cores
- PBIST mechanism for all the memories
- Error Correction Code (ECC) on the critical memories
- Memory Protection Unit (MPU) on all critical resources
- Voltage monitor on core supply
- DCC Clock monitors to monitor all the primary clocks. One Watchdog for APPSS core
- PLL Lock monitors – PHASELOCK, FREQLOCK
- Four temperature sensors to monitor internal temperature at temperature sensitive locations.
- One Error Signaling Modules (ESM) to enable error monitoring
- Dedicated hardware Memory Cyclic Redundancy Check (MCRC) blocks.

1.2 Device Block Diagram

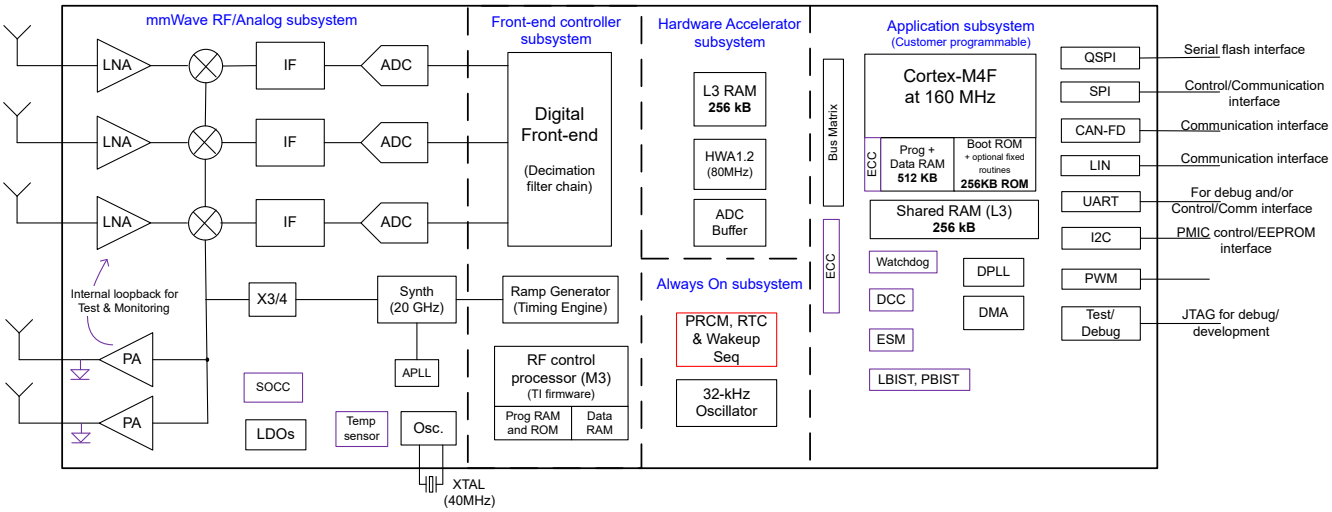


Figure 1-1. Device Block Diagram

ADVANCE INFORMATION

2 Memory Map

Table 2-1. xWRLx432 Main Memory map

Device Variant	APPSS_RAM	RADAR_DATACUBE_RAM	SHARED_RAM (shared between APPSS_RAM and RADAR_DATACUBE_RAM)	FECSS_RAM
xWRLx432	512 KB	256 KB	256 KB	32 KB

The device memory can be used in three different memory configurations as mentioned in Table 2-2.

Table 2-2. xWRLx432 Memory Usage

Device Memory Usage	APPSS_RAM	RADAR_DATACUBE_RAM	Calculation
Usage 1	768	256	APPSS_RAM uses 512 KB of APPSS_RAM+ 256 KB OF SHARED RAM, RADAR_DATACUBE_RAM uses 256KB of DATACUBE_RAM
Usage 2	640	384	APPSS_RAM uses 512 KB of APPSS_RAM+ 128 KB OF SHARED RAM, RADAR_DATACUBE_RAM uses 256KB of DATACUBE_RAM+ 128 KB OF SHARED RAM
Usage 3	512	512	APPSS_RAM uses 512 KB of APPSS_RAM, RADAR_DATACUBE_RAM uses 256KB of DATACUBE_RAM++ 256 KB OF SHARED RAM

APPSS RAM and Shared RAM consists of different memory banks as mentioned in the Table 2-3.

Table 2-3.

Memory Bank	Device Variant 1 (in KB)
Bank 1	256
Bank 2	128
Bank 3	128
Shared_RAM_1	128
Shared_RAM_2	128

For the APPSS RAM, addresses 0x00458000 to 0x00460000 should never be initialized with data prior to boot up since this region is used by the ROM bootloader. If optimizing for maximum memory usage, the stack and heap may be assigned to this memory region in the linker file since the stack and heap are initialized after booting. For more information on the boot up sequence, see the Device Initialization section.

2.1 Main Subsystem APPSS Cortex M4 Memory Map

Module Name[1]	Base Address	Size
APP_CPU_ROM	0x0000 0000	256 KBytes
APP_CPU_RAM	0x0040 0000	512 KBytes
APP_CPU_SHARED_RAM	0x0048 0000	256 KBytes
APP_CPU_QSPI	0x1000 0000	4 MBytes
FEC_IPC_RAM	0x2120 0000	2 KBytes
FEC_SHARED_RAM	0x2120 8000	96 KBytes
FEC_TIMING_RAM	0x2188 0000	8 KBytes
APP_ROM	0x2200 0000	256 KBytes

Module Name[1]	Base Address	Size
APP_RAM	0x2240 0000	512 KBytes
APP_SHARED_RAM	0x2248 0000	256 KBytes
FEC_LSTC	0x5204 0000	4 KBytes
FEC_PCR1	0x52F7 8000	1 KBytes
FEC_ECC_AGG	0x52F7 F800	528Bytes
APP_LIN	0x5300 0000	148Bytes
APP_CAN_MSG_RAM	0x5302 0000	68 KBytes
APP_PCR4	0x53F7 8000	1 KBytes
APP_UART_0	0x53F7 F000	148Bytes
APP_SPI_0	0x53F7 F400	420Bytes
APP_CANCFG	0x53F7 F800	768Bytes
APP_CANECC	0x53F7 FC00	528Bytes
TPTC_A0	0x5400 0000	860Bytes
TPTC_A1	0x5401 0000	860Bytes
APP_CRC	0x5402 0000	328Bytes
HWA_DMA0	0x5500 0000	32 KBytes
HWA_DMA1	0x5500 8000	32 KBytes
HWA_CFG	0x5501 0000	984Bytes
HWA_PARAM_MEM	0x5501 4000	1 KBytes
HWA_WINDOW_RAM	0x5501 8000	4 KBytes
HWA_MC_PING_RAM	0x5501 A000	2 KBytes
HWA_MC_PONG_RAM	0x5501 C000	2 KBytes
TPTC_B0	0x5502 0000	860Bytes
TPTC_B1	0x5504 0000	860Bytes
APP_HWA_ADCBUF_RD	0x5506 0000	16 KBytes
APP_HWA_ADCBUF_WR	0x5507 0000	16 KBytes
TPCC_B	0x5508 0000	16 KBytes
HWA_PCR	0x55F7 8000	1 KBytes
TPCC_A	0x5600 0000	16 KBytes
APP_RAM_2KB	0x5602 0000	2 KBytes
APP_RCM	0x5604 0000	4 KBytes
APP_CTRL	0x5606 0000	4 KBytes
APP_HWA_ADCBUF_CTRL	0x5608 0000	4 KBytes
APP_LSTC	0x560A 0000	4 KBytes
APP_PCR3	0x56F7 8000	1 KBytes
APP_ECC_AGG	0x56F7 EC00	528Bytes
APP_RTI	0x56F7 F000	192Bytes
APP_WD	0x56F7 F400	192Bytes
APP_DCC	0x56F7 F800	60Bytes
APP_ESM	0x56F7 FC00	220Bytes
APP_PCR6	0x57F7 8000	1 KBytes
APP_UART_1	0x57F7 F000	148Bytes
APP_SPI_1	0x57F7 F400	420Bytes
APP_I2C	0x57F7 F800	100Bytes
APP_PWM	0x57F7 FC00	116Bytes
APP_IDALLOC	0x5800 0000	4 KBytes
APP_MPU	0x5802 0000	780Bytes

Module Name[1]	Base Address	Size
FEC_MPU	0x5804 0000	780Bytes
APP_PCR5	0x58F7 8000	1 KBytes
TOP_IO_MUX	0x5A00 0000	112Bytes
TOP_PRCM	0x5A04 0000	7 KBytes
TOP_PCR2	0x5AF7 8000	1 KBytes
TOP_GIO	0x5AF7 FC00	340Bytes
TOPSS_CTRL	0x5B02 0000	4 KBytes
PLLDIG_CTRL	0x5B04 0000	4 KBytes
TOP_PCR8	0x5BF7 8000	1 KBytes
TOP_PBIST	0x5C02 0000	464Bytes
TOP_DEBUGSS	0x5CA0 0000	72 KBytes
TOP_PCR7	0x5CF7 8000	1 KBytes
APP_HWA_RAM	0x6000 0000	512 KBytes
APP_QSPI	0x7000 0000	4 MBytes
APP_CFG_QSPI	0x7800 0000	116Bytes

(1) When APP ROM and RAM memories are accessed from the APP core, the memory addresses of APP_CPU_ROM/APP_CPU_RAM can be used. When the memories are accessed from the other cores, the memory addresses of APP_ROM and APP_RAM can be used.

3 System Interconnect

The device implements a system interconnect based on TI's common bus architecture, comprising of VBUSM and VBUSP protocols, AHB Interconnect Matrix and bridges for protocol, width and frequency conversions.

The system interconnect is designed for the high-performance needs of the system. The interconnect structure is a full crossbar implementation, wherein every controller has an independent communication path with every target such that transactions from each controllers have access to full interconnect bandwidth. Arbitration only happens at taget end point.

The following terms used in the context of connections within a subsystem-

- SCR - Switched Central Resource
- Infrastructure - To refer to combined entity of - (i) VBUSP/VBUSM SCR (ii) AHB Matrix (iii) bridges for protocol conversion

Table 3-1 shows the possible access among subsystems. Each of these subsystems are asynchronous to each other, and has VBUSP bridges added between the subsystems. HWASS and APPSS have synchronous access.

Table 3-1. Subsystem Access

Target->	APPSS	FECSS	TOPSS	HWASS
APPSS	N/A	Direct (async)	Direct (async)	Direct (sync)
FECSS	Direct (async)	N/A	(via APPSS)	(via APPSS)
TOPSS	Direct (async)	(via APPSS)	N/A	(via APPSS)
HWASS	Direct (sync)	(via APPSS)	No access	N/A

3.1 APPSS Infrastructure

In the APP subsystem, the primary VBUSP-SCR is responsible for managing the arbitration priority between accesses from multiple controllers to each of the targets. The arbitration priority is always round-robin.

The APP subsystem has PCR interconnect that manages the accesses to the peripheral registers and peripheral memories, and provides a global reset for all peripherals. 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.

The APPSS has the following controllers on its' SCR:

- CortexM4
- TPTC_A0
- TPTC_A1

The following are the subsystems that are controllers on the APPSS SCR:

- HWA
- TOPSS
- FECSS

The controller CortexM4 is connected to the CPU memories through AHB interconnect-matrix.

The ratio between the CPU, CPU Memories, AHB Matrix and VBUSP-SCR is always 2:1. Cortex-M4 , AHB interconnect-matrix and CPU memories operate at a maximum of 160 MHz, while the rest of the APPSS operates at maximum 80 MHz including VBUSP SCR and the peripherals.

APPSS can access peripherals in HWASS/FECSS/TOPSS. Since HWASS and FECSS can be powered down when the APPSS is still active, when either FECSS/HWASS is powered down, the access to FECSS and HWASS is blocked and an ABORT is generated.

The following diagram illustrates the connection at a absratced level.

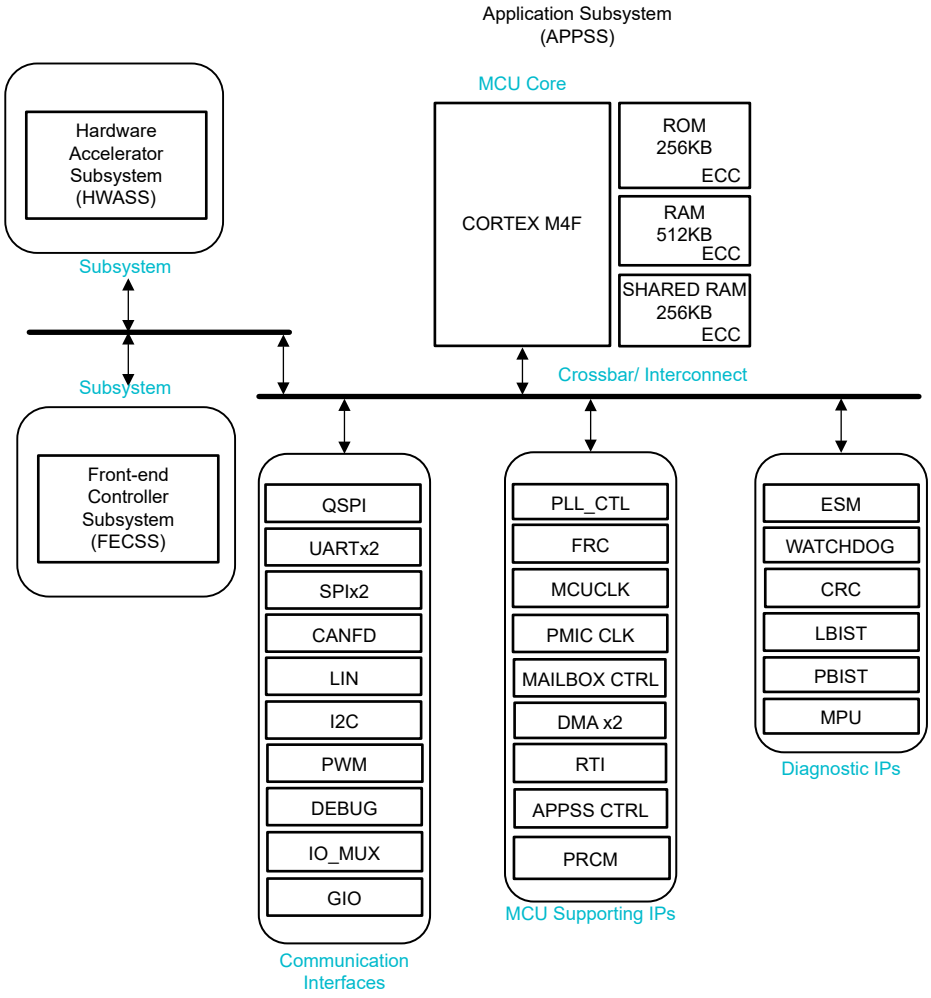


Figure 3-1. Application Subsystem Infrastructure

3.2 FECSS Infrastructure

The FECSS Infrastructure has the following:

- AHB Matrix
- VBUSP SCR
- AHB2VBUSP and VBUSP2AHB for protocol conversion

The FECSS has the following controllers on it's AHB Matrix:

- CortexM3

The following are the subsystems that are controllers on the FECSS SCR:

- APPSS

The controller CortexM3 is connected to the CPU memories through AHB interconnect-matrix and VBUSP-SCR and peripherals operates with the same clock of maximum frequency 80 MHz. Unlike APPSS there is no difference in frequency of operation between CPU and VBUSP-SCR in FECSS.

The application developers can access FEC_IPC_RAM and FEC_TIMING_RAM from APPSS. For more information, refer to mmwave DFP APIs.

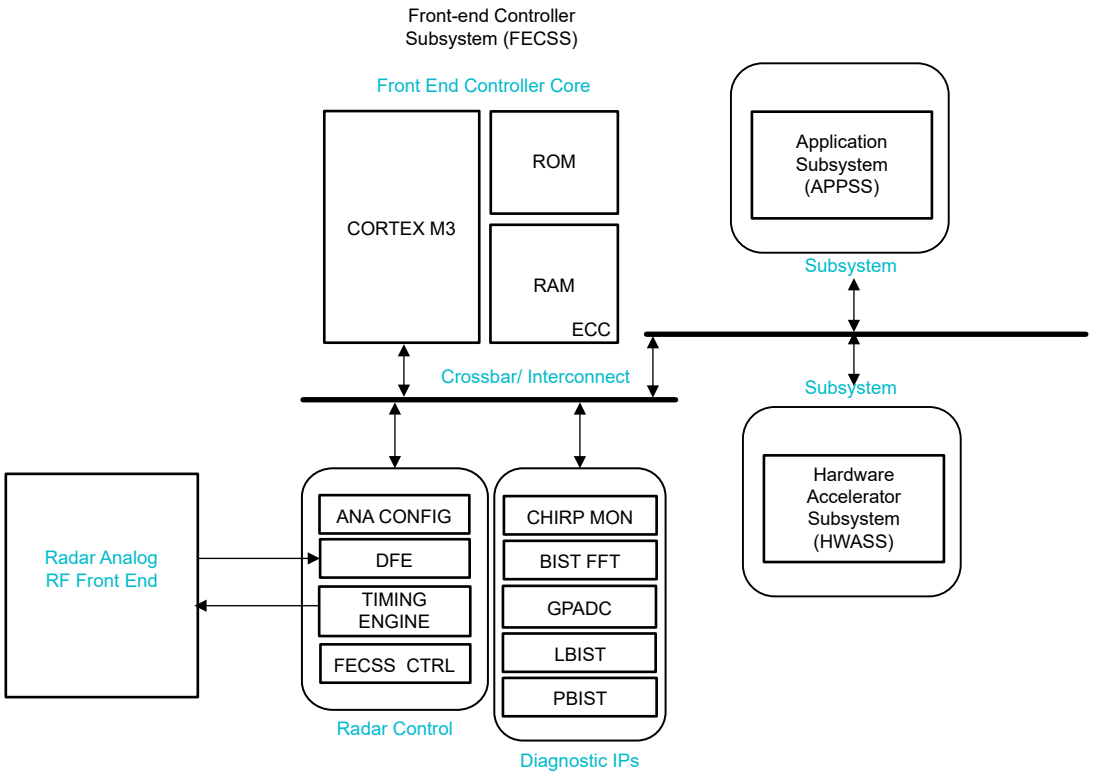


Figure 3-2. Front-End Controller Subsystem Infrastructure

3.3 HWASS Infrastructure

The HWASS has a 64-bit data SCR.

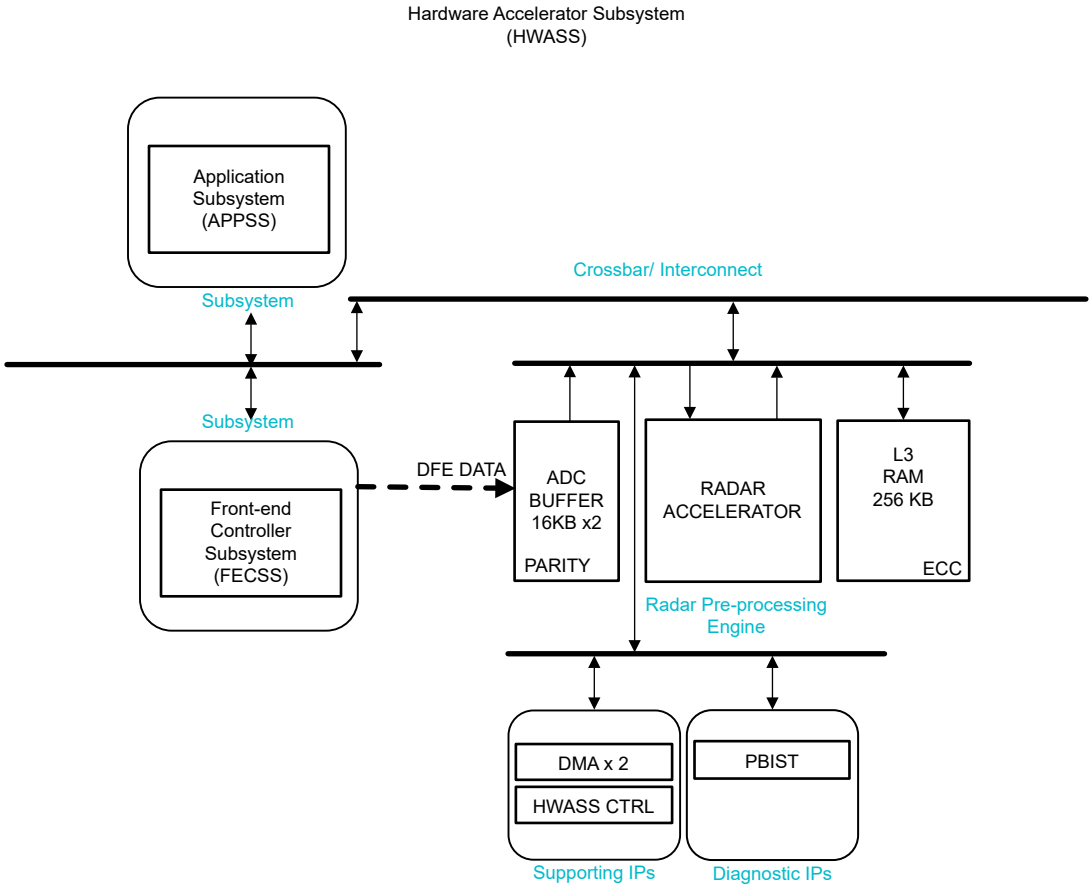


Figure 3-3. Hardware Accelerator Subsystem Infrastructure

3.4 Peripheral Central Resource

Peripheral Central Resource (PCR)

There are a total of 11 PCRs in the device.

- APP_PCR3
- APP_PCR4
- APP_PCR5
- APP_PCR6
- FEC_PCR1
- FEC_PCR11
- FEC_PCR12
- TOP_PCR2
- TOP_PCR7
- TOP_PCR8

Table 3-2. APP_PCR3 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	TPCC_A	PCS0	NONE	Yes	NO
2	APP_RAM_2KB	PCS1	NONE	No	Tied to 0
3	APP_RCM	PCS2	NONE	Yes	Tied to 0
4	APP_CTRL	PCS3	NONE	Yes	Tied to 0
5	APP_HWA_ADCBUF_CTRL	PCS4	NONE	Yes	Tied to 0
6	APP_LSTC	PCS5	NONE	No	Tied to 0

Table 3-2. APP_PCR3 Mapping to Target (continued)

7	APP_PCR3	PS30,31	All Quadrants	No	Tied to 0
8	APP_ECC_AGG	PS4	0,1,2,3	No	Tied to 0
9	APP_RTI	PS3	0	No	Tied to 0
10	APP_WD	PS2	0	No	Tied to 0
11	APP_DCC	PS1	0	No	Tied to 0
12	APP_ESM	PS0	0,1	No	Tied to 0

Table 3-3. APP_PCR4 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	APP_LIN	PCS0	NONE	No	Tied to 0
2	APP_CAN_MSG_RAM	PCS1	NONE	No	Tied to 0
3	APP_PCR4	PS30,31	All Quadrants	No	Tied to 0
4	APP_UART_0	PS3	0,1,2,3	No	Tied to 0
5	APP_SPI_0	PS2	0,1	No	Tied to 0
6	APP_CANCFG	PS1	0,1,2,3	No	Tied to 0
7	APP_CANECC	PS0	0,1,2,3	No	Tied to 0

Table 3-4. APP_PCR5 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	APP_IDALLOC	PCS0	NONE	Yes	Unconnected (X is seen)
2	APP_MPU	PCS1	NONE	Yes	Tied to 0
3	FEC_MPU	PCS2	NONE	Yes	Tied to 0
4	APP_PCR5	PS30,31	All Quadrants	No	Tied to 0

Table 3-5. APP_PCR6 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	APP_PCR6	PS30,31	All Quadrants	No	Tied to 0
2	APP_UART_1	PS3	0,1,2,3	No	Tied to 0
3	APP_SPI_1	PS2	0,1	No	Tied to 0
4	APP_I2C	PS1	0,1	No	Tied to 0
5	APP_PWM	PS0	0	No	Tied to 0

Table 3-6. FEC_PCR1 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
3	FEC_LSTC	PCS2	NONE	No	Tied to 0
4	FEC_PCR1	PS30,31	All Quadrants	No	Tied to 0
5	FEC_ECC_AGG	PS1	0,1,2,3	Yes	Tied to 0

Table 3-7. TOP_PCR2 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	TOP_IO_MUX	PCS0	NONE	Yes	NO
2	TOP_EFUSE	PCS1	NONE	Yes	Tied to 0
3	TOP_PRCM	PCS2	NONE	Yes	Tied to 0
4	TOP_PCR2	PS30,31	All Quadrants	No	Tied to 0
5	TOP_GIO	PS0	0,1	Yes	Tied to 0

ADVANCE INFORMATION

Table 3-8. TOP_PCR7 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	TOP_PBIST	PCS1	NONE	No	NO
2	EFUSE_FARM	PCS3	NONE	Yes	Tied to 0
3	TOP_PCR7	PS30,31	All Quadrants	No	Tied to 0

Table 3-9. TOP_PCR8 Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	FRAME_TIMER	PCS0	NONE	Yes	Tied to 0
2	TOPSS_CTRL	PCS1	NONE	Yes	Tied to 0
3	PLLDIG_CTRL	PCS2	NONE	Yes	Tied to 0
4	APLL_CTRL	PCS3	NONE	Yes	Tied to 0
5	TOP_PCR8	PS30,31	All Quadrants	No	Tied to 0

Table 3-10. HWA_PCR Mapping to Target

Sl No	Peripheral Name	PCR Region	Quadrants	Address ERROR(Yes/No)	PCR_AERROR (TIED TO 0/NO)
1	TPTC_B0	PCS1	NONE	Yes	NO
2	TPTC_B1	PCS2	NONE	Yes	NO
3	TPCC_B	PCS4	NONE	Yes	NO
4	HWA_PCR	PS30,31	All Quadrants	No	Tied to 0

PCR Base Address

Module Name	Base Address	Size
FEC_PCR12	0x50F7 8000	1 KBytes
FEC_PCR11	0x51F7 8000	1 KBytes
FEC_PCR1	0x52F7 8000	1 KBytes
APP_PCR4	0x53F7 8000	1 KBytes
HWA_PCR	0x55F7 8000	1 KBytes
APP_PCR3	0x56F7 8000	1 KBytes
APP_PCR6	0x57F7 8000	1 KBytes
APP_PCR5	0x58F7 8000	1 KBytes
TOP_PCR2	0x5AF7 8000	1 KBytes
TOP_PCR8	0x5BF7 8000	1 KBytes
TOP_PCR7	0x5CF7 8000	1 KBytes

3.4.1 Subsystem_PCR Registers

Table 3-11 lists the memory-mapped registers for the Subsystem_PCR registers. All register offset addresses not listed in Table 3-11 should be considered as reserved locations and the register contents should not be modified.

Table 3-11. Subsystem_PCR Registers

Offset	Acronym	Register Name	Section
0h	PMPROTSET0	PMPROTSET0	PMPROTSET0 Register (Offset = 0h) [Reset = 00000000h]
4h	PMPROTSET1	PMPROTSET1	PMPROTSET1 Register (Offset = 4h) [Reset = 00000000h]
10h	PMPROTCLR0	PMPROTCLR0	PMPROTCLR0 Register (Offset = 10h) [Reset = 00000000h]
14h	PMPROTCLR1	PMPROTCLR1	PMPROTCLR1 Register (Offset = 14h) [Reset = 00000000h]
20h	PPROTSET_0	PPROTSET_0	PPROTSET_0 Register (Offset = 20h) [Reset = 00000000h]
24h	PPROTSET_1	PPROTSET_1	PPROTSET_1 Register (Offset = 24h) [Reset = 00000000h]
28h	PPROTSET_2	PPROTSET_2	PPROTSET_2 Register (Offset = 28h) [Reset = 00000000h]
2Ch	PPROTSET_3	PPROTSET_3	PPROTSET_3 Register (Offset = 2Ch) [Reset = 00000000h]
40h	PPROTCLR0	PPROTCLR0	PPROTCLR0 Register (Offset = 40h) [Reset = 00000000h]
44h	PPROTCLR1	PPROTCLR1	PPROTCLR1 Register (Offset = 44h) [Reset = 00000000h]
48h	PPROTCLR2	PPROTCLR2	PPROTCLR2 Register (Offset = 48h) [Reset = 00000000h]
4Ch	PPROTCLR3	PPROTCLR3	PPROTCLR3 Register (Offset = 4Ch) [Reset = 00000000h]
60h	PCSPWRDWNSET0	PCSPWRDWNSET0	PCSPWRDWNSET0 Register (Offset = 60h) [Reset = 0000001Fh]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
64h	PCSPWRDWNSET1	PCSPWRDWNSET1	PCSPWRDWNSET1 Register (Offset = 64h) [Reset = 00000000h]
70h	PCSPWRDWNCLR0	PCSPWRDWNCLR0	PCSPWRDWNCLR 0 Register (Offset = 70h) [Reset = 0000001Fh]
74h	PCSPWRDWNCLR1	PCSPWRDWNCLR1	PCSPWRDWNCLR 1 Register (Offset = 74h) [Reset = 00000000h]
80h	PSPWRDWNSET0	PSPWRDWNSET0	PSPWRDWNSET0 Register (Offset = 80h) [Reset = 00011111h]
84h	PSPWRDWNSET1	PSPWRDWNSET1	PSPWRDWNSET1 Register (Offset = 84h) [Reset = 00000000h]
88h	PSPWRDWNSET2	PSPWRDWNSET2	PSPWRDWNSET2 Register (Offset = 88h) [Reset = 00000000h]
8Ch	PSPWRDWNSET3	PSPWRDWNSET3	PSPWRDWNSET3 Register (Offset = 8Ch) [Reset = 00000000h]
A0h	PSPWRDWNCLR0	PSPWRDWNCLR0	PSPWRDWNCLR0 Register (Offset = A0h) [Reset = 00011111h]
A4h	PSPWRDWNCLR1	PSPWRDWNCLR1	PSPWRDWNCLR1 Register (Offset = A4h) [Reset = 00000000h]
A8h	PSPWRDWNCLR2	PSPWRDWNCLR2	PSPWRDWNCLR2 Register (Offset = A8h) [Reset = 00000000h]
ACh	PSPWRDWNCLR3	PSPWRDWNCLR3	PSPWRDWNCLR3 Register (Offset = ACh) [Reset = 00000000h]
C0h	PDPWRDWNSET	PDPWRDWNSET	PDPWRDWNSET Register (Offset = C0h) [Reset = 00000001h]
C4h	PDPWRDWNCLR	PDPWRDWNCLR	PDPWRDWNCLR Register (Offset = C4h) [Reset = 00000001h]
200h	MSTIDWRENA	MSTIDWRENA	MSTIDWRENA Register (Offset = 200h) [Reset = 00000005h]
204h	MSTIDENA	MSTIDENA	MSTIDENA Register (Offset = 204h) [Reset = 00000005h]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
208h	MSTIDDIAGCTRL	MSTIDDIAGCTRL	MSTIDDIAGCTRL Register (Offset = 208h) [Reset = 00000005h]
300h	PS0MSTID_L	PS0MSTID_L	PS0MSTID_L Register (Offset = 300h) [Reset = 0000FFFFh]
304h	PS0MSTID_H	PS0MSTID_H	PS0MSTID_H Register (Offset = 304h) [Reset = 00000000h]
308h	PS1MSTID_L	PS1MSTID_L	PS1MSTID_L Register (Offset = 308h) [Reset = 0000FFFFh]
30Ch	PS1MSTID_H	PS1MSTID_H	PS1MSTID_H Register (Offset = 30Ch) [Reset = 00000000h]
310h	PS2MSTID_L	PS2MSTID_L	PS2MSTID_L Register (Offset = 310h) [Reset = 0000FFFFh]
314h	PS2MSTID_H	PS2MSTID_H	PS2MSTID_H Register (Offset = 314h) [Reset = 00000000h]
318h	PS3MSTID_L	PS3MSTID_L	PS3MSTID_L Register (Offset = 318h) [Reset = 0000FFFFh]
31Ch	PS3MSTID_H	PS3MSTID_H	PS3MSTID_H Register (Offset = 31Ch) [Reset = 00000000h]
320h	PS4MSTID_L	PS4MSTID_L	PS4MSTID_L Register (Offset = 320h) [Reset = 0000FFFFh]
324h	PS4MSTID_H	PS4MSTID_H	PS4MSTID_H Register (Offset = 324h) [Reset = 00000000h]
328h	PS5MSTID_L	PS5MSTID_L	PS5MSTID_L Register (Offset = 328h) [Reset = 00000000h]
32Ch	PS5MSTID_H	PS5MSTID_H	PS5MSTID_H Register (Offset = 32Ch) [Reset = 00000000h]
330h	PS6MSTID_L	PS6MSTID_L	PS6MSTID_L Register (Offset = 330h) [Reset = 00000000h]
334h	PS6MSTID_H	PS6MSTID_H	PS6MSTID_H Register (Offset = 334h) [Reset = 00000000h]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
338h	PS7MSTID_L	PS7MSTID_L	PS7MSTID_L Register (Offset = 338h) [Reset = 00000000h]
33Ch	PS7MSTID_H	PS7MSTID_H	PS7MSTID_H Register (Offset = 33Ch) [Reset = 00000000h]
340h	PS8MSTID_L	PS8MSTID_L	PS8MSTID_L Register (Offset = 340h) [Reset = 00000000h]
344h	PS8MSTID_H	PS8MSTID_H	PS8MSTID_H Register (Offset = 344h) [Reset = 00000000h]
348h	PS9MSTID_L	PS9MSTID_L	PS9MSTID_L Register (Offset = 348h) [Reset = 00000000h]
34Ch	PS9MSTID_H	PS9MSTID_H	PS9MSTID_H Register (Offset = 34Ch) [Reset = 00000000h]
350h	PS10MSTID_L	PS10MSTID_L	PS10MSTID_L Register (Offset = 350h) [Reset = 00000000h]
354h	PS10MSTID_H	PS10MSTID_H	PS10MSTID_H Register (Offset = 354h) [Reset = 00000000h]
358h	PS11MSTID_L	PS11MSTID_L	PS11MSTID_L Register (Offset = 358h) [Reset = 00000000h]
35Ch	PS11MSTID_H	PS11MSTID_H	PS11MSTID_H Register (Offset = 35Ch) [Reset = 00000000h]
360h	PS12MSTID_L	PS12MSTID_L	PS12MSTID_L Register (Offset = 360h) [Reset = 00000000h]
364h	PS12MSTID_H	PS12MSTID_H	PS12MSTID_H Register (Offset = 364h) [Reset = 00000000h]
368h	PS13MSTID_L	PS13MSTID_L	PS13MSTID_L Register (Offset = 368h) [Reset = 00000000h]
36Ch	PS13MSTID_H	PS13MSTID_H	PS13MSTID_H Register (Offset = 36Ch) [Reset = 00000000h]
370h	PS14MSTID_L	PS14MSTID_L	PS14MSTID_L Register (Offset = 370h) [Reset = 00000000h]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
374h	PS14MSTID_H	PS14MSTID_H	PS14MSTID_H Register (Offset = 374h) [Reset = 00000000h]
378h	PS15MSTID_L	PS15MSTID_L	PS15MSTID_L Register (Offset = 378h) [Reset = 00000000h]
37Ch	PS15MSTID_H	PS15MSTID_H	PS15MSTID_H Register (Offset = 37Ch) [Reset = 00000000h]
380h	PS16MSTID_L	PS16MSTID_L	PS16MSTID_L Register (Offset = 380h) [Reset = 00000000h]
384h	PS16MSTID_H	PS16MSTID_H	PS16MSTID_H Register (Offset = 384h) [Reset = 00000000h]
388h	PS17MSTID_L	PS17MSTID_L	PS17MSTID_L Register (Offset = 388h) [Reset = 00000000h]
38Ch	PS17MSTID_H	PS17MSTID_H	PS17MSTID_H Register (Offset = 38Ch) [Reset = 00000000h]
390h	PS18MSTID_L	PS18MSTID_L	PS18MSTID_L Register (Offset = 390h) [Reset = 00000000h]
394h	PS18MSTID_H	PS18MSTID_H	PS18MSTID_H Register (Offset = 394h) [Reset = 00000000h]
398h	PS19MSTID_L	PS19MSTID_L	PS19MSTID_L Register (Offset = 398h) [Reset = 00000000h]
39Ch	PS19MSTID_H	PS19MSTID_H	PS19MSTID_H Register (Offset = 39Ch) [Reset = 00000000h]
3A0h	PS20MSTID_L	PS20MSTID_L	PS20MSTID_L Register (Offset = 3A0h) [Reset = 00000000h]
3A4h	PS20MSTID_H	PS20MSTID_H	PS20MSTID_H Register (Offset = 3A4h) [Reset = 00000000h]
3A8h	PS21MSTID_L	PS21MSTID_L	PS21MSTID_L Register (Offset = 3A8h) [Reset = 00000000h]
3ACh	PS21MSTID_H	PS21MSTID_H	PS21MSTID_H Register (Offset = 3ACh) [Reset = 00000000h]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
3B0h	PS22MSTID_L	PS22MSTID_L	PS22MSTID_L Register (Offset = 3B0h) [Reset = 00000000h]
3B4h	PS22MSTID_H	PS22MSTID_H	PS22MSTID_H Register (Offset = 3B4h) [Reset = 00000000h]
3B8h	PS23MSTID_L	PS23MSTID_L	PS23MSTID_L Register (Offset = 3B8h) [Reset = 00000000h]
3BCh	PS23MSTID_H	PS23MSTID_H	PS23MSTID_H Register (Offset = 3BCh) [Reset = 00000000h]
3C0h	PS24MSTID_L	PS24MSTID_L	PS24MSTID_L Register (Offset = 3C0h) [Reset = 00000000h]
3C4h	PS24MSTID_H	PS24MSTID_H	PS24MSTID_H Register (Offset = 3C4h) [Reset = 00000000h]
3C8h	PS25MSTID_L	PS25MSTID_L	PS25MSTID_L Register (Offset = 3C8h) [Reset = 00000000h]
3CCh	PS25MSTID_H	PS25MSTID_H	PS25MSTID_H Register (Offset = 3CCh) [Reset = 00000000h]
3D0h	PS26MSTID_L	PS26MSTID_L	PS26MSTID_L Register (Offset = 3D0h) [Reset = 00000000h]
3D4h	PS26MSTID_H	PS26MSTID_H	PS26MSTID_H Register (Offset = 3D4h) [Reset = 00000000h]
3D8h	PS27MSTID_L	PS27MSTID_L	PS27MSTID_L Register (Offset = 3D8h) [Reset = 00000000h]
3DCh	PS27MSTID_H	PS27MSTID_H	PS27MSTID_H Register (Offset = 3DCh) [Reset = 00000000h]
3E0h	PS28MSTID_L	PS28MSTID_L	PS28MSTID_L Register (Offset = 3E0h) [Reset = 00000000h]
3E4h	PS28MSTID_H	PS28MSTID_H	PS28MSTID_H Register (Offset = 3E4h) [Reset = 00000000h]
3E8h	PS29MSTID_L	PS29MSTID_L	PS29MSTID_L Register (Offset = 3E8h) [Reset = 00000000h]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
3ECh	PS29MSTID_H	PS29MSTID_H	PS29MSTID_H Register (Offset = 3ECh) [Reset = 00000000h]
3F0h	PS30MSTID_L	PS30MSTID_L	PS30MSTID_L Register (Offset = 3F0h) [Reset = 0000FFFFh]
3F4h	PS30MSTID_H	PS30MSTID_H	PS30MSTID_H Register (Offset = 3F4h) [Reset = 00000000h]
3F8h	PS31MSTID_L	PS31MSTID_L	PS31MSTID_L Register (Offset = 3F8h) [Reset = 00000000h]
3FCh	PS31MSTID_H	PS31MSTID_H	PS31MSTID_H Register (Offset = 3FCh) [Reset = 00000000h]
540h	PCS0MSTID	PCS0MSTID	PCS0MSTID Register (Offset = 540h) [Reset = FFFFFFFFh]
544h	PCS1MSTID	PCS1MSTID	PCS1MSTID Register (Offset = 544h) [Reset = FFFFFFFFh]
548h	PCS2MSTID	PCS2MSTID	PCS2MSTID Register (Offset = 548h) [Reset = 0000FFFFh]
54Ch	PCS3MSTID	PCS3MSTID	PCS3MSTID Register (Offset = 54Ch) [Reset = 00000000h]
550h	PCS4MSTID	PCS4MSTID	PCS4MSTID Register (Offset = 550h) [Reset = 00000000h]
554h	PCS5MSTID	PCS5MSTID	PCS5MSTID Register (Offset = 554h) [Reset = 00000000h]
558h	PCS6MSTID	PCS6MSTID	PCS6MSTID Register (Offset = 558h) [Reset = 00000000h]
55Ch	PCS7MSTID	PCS7MSTID	PCS7MSTID Register (Offset = 55Ch) [Reset = 00000000h]
560h	PCS8MSTID	PCS8MSTID	PCS8MSTID Register (Offset = 560h) [Reset = 00000000h]
564h	PCS9MSTID	PCS9MSTID	PCS9MSTID Register (Offset = 564h) [Reset = 00000000h]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
568h	PCS10MSTID	PCS10MSTID	PCS10MSTID Register (Offset = 568h) [Reset = 00000000h]
56Ch	PCS11MSTID	PCS11MSTID	PCS11MSTID Register (Offset = 56Ch) [Reset = 00000000h]
570h	PCS12MSTID	PCS12MSTID	PCS12MSTID Register (Offset = 570h) [Reset = 00000000h]
574h	PCS13MSTID	PCS13MSTID	PCS13MSTID Register (Offset = 574h) [Reset = 00000000h]
578h	PCS14MSTID	PCS14MSTID	PCS14MSTID Register (Offset = 578h) [Reset = 00000000h]
57Ch	PCS15MSTID	PCS15MSTID	PCS15MSTID Register (Offset = 57Ch) [Reset = 00000000h]
580h	PCS16MSTID	PCS16MSTID	PCS16MSTID Register (Offset = 580h) [Reset = 00000000h]
584h	PCS17MSTID	PCS17MSTID	PCS17MSTID Register (Offset = 584h) [Reset = 00000000h]
588h	PCS18MSTID	PCS18MSTID	PCS18MSTID Register (Offset = 588h) [Reset = 00000000h]
58Ch	PCS19MSTID	PCS19MSTID	PCS19MSTID Register (Offset = 58Ch) [Reset = 00000000h]
590h	PCS20MSTID	PCS20MSTID	PCS20MSTID Register (Offset = 590h) [Reset = 00000000h]
594h	PCS21MSTID	PCS21MSTID	PCS21MSTID Register (Offset = 594h) [Reset = 00000000h]
598h	PCS22MSTID	PCS22MSTID	PCS22MSTID Register (Offset = 598h) [Reset = 00000000h]
59Ch	PCS23MSTID	PCS23MSTID	PCS23MSTID Register (Offset = 59Ch) [Reset = 00000000h]
5A0h	PCS24MSTID	PCS24MSTID	PCS24MSTID Register (Offset = 5A0h) [Reset = 00000000h]

Table 3-11. Subsystem_PCR Registers (continued)

Offset	Acronym	Register Name	Section
5A4h	PCS25MSTID	PCS25MSTID	PCS25MSTID Register (Offset = 5A4h) [Reset = 00000000h]
5A8h	PCS26MSTID	PCS26MSTID	PCS26MSTID Register (Offset = 5A8h) [Reset = 00000000h]
5ACh	PCS27MSTID	PCS27MSTID	PCS27MSTID Register (Offset = 5ACh) [Reset = 00000000h]
5B0h	PCS28MSTID	PCS28MSTID	PCS28MSTID Register (Offset = 5B0h) [Reset = 00000000h]
5B4h	PCS29MSTID	PCS29MSTID	PCS29MSTID Register (Offset = 5B4h) [Reset = 00000000h]
5B8h	PCS30MSTID	PCS30MSTID	PCS30MSTID Register (Offset = 5B8h) [Reset = 00000000h]
5BCh	PCS31MSTID	PCS31MSTID	PCS31MSTID Register (Offset = 5BCh) [Reset = 00000000h]
5E0h	PCREXTMSTID	PCREXTMSTID	PCREXTMSTID Register (Offset = 5E0h) [Reset = 00000000h]

Complex bit access types are encoded to fit into small table cells. Table 3-12 shows the codes that are used for access types in this section.

Table 3-12. Subsystem_PCR 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

3.4.1.1 PMPROTSET0 Register (Offset = 0h) [Reset = 00000000h]

PMPROTSET0 is shown in Table 3-13.

Return to the Summary Table.

Set-only register to protect PCS frames 0 to 31

Table 3-13. PMPROTSET0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS31_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
30	PCS30_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
29	PCS29_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
28	PCS28_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
27	PCS27_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
26	PCS26_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-13. PMPROTSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	PCS25_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
24	PCS24_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
23	PCS23_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
22	PCS22_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
21	PCS21_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
20	PCS20_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-13. PMPROTSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	PCS19_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
18	PCS18_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
17	PCS17_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
16	PCS16_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
15	PCS15_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
14	PCS14_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-13. PMPROTSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	PCS13_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
12	PCS12_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
11	PCS11_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
10	PCS10_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
9	PCS9_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
8	PCS8_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-13. PMPROTSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	PCS7_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
6	PCS6_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
5	PCS5_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
4	PCS4_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
3	PCS3_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
2	PCS2_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-13. PMPROTSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PCS1_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
0	PCS0_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a peripheral at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.2 PMPROTSET1 Register (Offset = 4h) [Reset = 00000000h]

PMPROTSET1 is shown in Table 3-14.

Return to the Summary Table.

Set-only register to protect PCS frames 32 to 63

Table 3-14. PMPROTSET1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS63_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
30	PCS62_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
29	PCS61_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
28	PCS60_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

Table 3-14. PMPROTSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	PCS59_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
26	PCS58_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
25	PCS57_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
24	PCS56_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
23	PCS55_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
22	PCS54_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
21	PCS53_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
20	PCS52_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
19	PCS51_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

Table 3-14. PMPROTSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	PCS50_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
17	PCS49_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
16	PCS48_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
15	PCS47_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
14	PCS46_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
13	PCS45_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
12	PCS44_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
11	PCS43_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
10	PCS42_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

Table 3-14. PMPROTSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	PCS41_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
8	PCS40_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
7	PCS39_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
6	PCS38_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
5	PCS37_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
4	PCS36_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
3	PCS35_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
2	PCS34_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
1	PCS33_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

Table 3-14. PMPROTSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PCS32_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

3.4.1.3 PMPROTCLR0 Register (Offset = 10h) [Reset = 00000000h]

PMPROTCLR0 is shown in Table 3-15.

Return to the Summary Table.

Clear-only register to protect PCS frames 0 to 31

Table 3-15. PMPROTCLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS31_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
30	PCS30_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
29	PCS29_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
28	PCS28_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
27	PCS27_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
26	PCS26_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

Table 3-15. PMPROTCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	PCS25_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
24	PCS24_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
23	PCS23_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
22	PCS22_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
21	PCS21_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
20	PCS20_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
19	PCS19_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
18	PCS18_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
17	PCS17_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

Table 3-15. PMPROTCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	PCS16_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
15	PCS15_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
14	PCS14_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
13	PCS13_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
12	PCS12_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
11	PCS11_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
10	PCS10_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
9	PCS9_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
8	PCS8_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

Table 3-15. PMPROTCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	PCS7_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
6	PCS6_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
5	PCS5_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
4	PCS4_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
3	PCS3_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
2	PCS2_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
1	PCS1_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
0	PCS0_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

3.4.1.4 PMPROTCLR1 Register (Offset = 14h) [Reset = 00000000h]

PMPROTCLR1 is shown in Table 3-16.

Return to the Summary Table.

Clear-only register to protect PCS frames 32 to 63

Table 3-16. PMPROTCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS63_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
30	PCS62_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
29	PCS61_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
28	PCS60_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
27	PCS59_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
26	PCS58_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
25	PCS57_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
24	PCS56_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

Table 3-16. PMPROTCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	PCS55_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
22	PCS54_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
21	PCS53_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
20	PCS52_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
19	PCS51_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
18	PCS50_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
17	PCS49_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
16	PCS48_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
15	PCS47_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

Table 3-16. PMPROTCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	PCS46_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
13	PCS45_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
12	PCS44_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
11	PCS43_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
10	PCS42_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
9	PCS41_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
8	PCS40_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
7	PCS39_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
6	PCS38_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

Table 3-16. PMPROTCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PCS37_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
4	PCS36_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
3	PCS35_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
2	PCS34_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
1	PCS33_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
0	PCS32_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

ADVANCE INFORMATION

3.4.1.5 PPROTSET_0 Register (Offset = 20h) [Reset = 00000000h]

PPROTSET_0 is shown in Table 3-17.

Return to the Summary Table.

Set-only register to protect the 32 quadrants of PS0 to PS7

Table 3-17. PPROTSET_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-17. PPROTSET_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
30	PS7_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
24	PS6_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-17. PPROTSET_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	PS5_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
16	PS4_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-17. PPROTSET_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	PS3_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
8	PS2_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-17. PPROTSET_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PS0_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
0	PS0_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

3.4.1.6 PPROTSET_1 Register (Offset = 24h) [Reset = 00000000h]

PPROTSET_1 is shown in Table 3-18.

Return to the Summary Table.

Set-only register to protect the 32 quadrants of PS8 to PS15

Table 3-18. PPROTSET_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-18. PPROTSET_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	PS15_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
24	PS14_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-18. PPROTSET_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	PS12_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
16	PS12_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-18. PPROTSET_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PS10_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
8	PS10_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-18. PPROTSET_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PS8_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
0	PS8_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

3.4.1.7 PPROTSET_2 Register (Offset = 28h) [Reset = 00000000h]

PPROTSET_2 is shown in Table 3-19.

Return to the Summary Table.

Set-only register to protect the 32 quadrants of PS16 to PS23

Table 3-19. PPROTSET_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

Table 3-19. PPROTSET_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	PS22_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
24	PS22_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

Table 3-19. PPROTSET_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PS20_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
16	PS20_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

Table 3-19. PPROTSET_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
0	PS16_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

3.4.1.8 PPROTSET_3 Register (Offset = 2Ch) [Reset = 00000000h]

PPROTSET_3 is shown in Table 3-20.

Return to the Summary Table.

Set-only register to protect the 32 quadrants of PS24 to PS31

Table 3-20. PPROTSET_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
24	PS30_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

Table 3-20. PPROTSET_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	PS29_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
16	PS28_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

Table 3-20. PPROTSET_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	PS27_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
8	PS26_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

Table 3-20. PPROTSET_3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PS25_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
0	PS24_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

ADVANCE INFORMATION

3.4.1.9 PPROTCLR0 Register (Offset = 40h) [Reset = 00000000h]

PPROTCLR0 is shown in Table 3-21.

Return to the Summary Table.

Clear-only register to protect the 32 quadrants of PS0 to PS7

Table 3-21. PPROTCLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-21. PPROTCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
30	PS7_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
24	PS6_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-21. PPROTCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	PS5_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
16	PS4_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-21. PPROTCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	PS3_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
8	PS2_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

Table 3-21. PPROTCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PS0_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
0	PS0_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

3.4.1.10 PPROTCLR1 Register (Offset = 44h) [Reset = 00000000h]

PPROTCLR1 is shown in Table 3-22.

Return to the Summary Table.

Clear-only register to protect the 32 quadrants of PS8 to PS15

Table 3-22. PPROTCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-22. PPROTCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	PS15_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
24	PS14_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-22. PPROTCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	PS12_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
16	PS12_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-22. PPROTCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PS10_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
8	PS10_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

Table 3-22. PPROTCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PS8_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
0	PS8_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

3.4.1.11 PPROTCLR2 Register (Offset = 48h) [Reset = 00000000h]

PPROTCLR2 is shown in Table 3-23.

Return to the Summary Table.

Clear-only register to protect the 32 quadrants of PS16 to PS23

Table 3-23. PPROTCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

Table 3-23. PPROTCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	PS22_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
24	PS22_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

Table 3-23. PPROTCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PS20_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
16	PS20_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

Table 3-23. PPROTCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
0	PS16_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

3.4.1.12 PPROTCLR3 Register (Offset = 4Ch) [Reset = 00000000h]

PPROTCLR3 is shown in Table 3-24.

Return to the Summary Table.

Clear-only register to protect the 32 quadrants of PS24 to PS31

Table 3-24. PPROTCLR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
24	PS30_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

Table 3-24. PPROTCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	PS29_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
16	PS28_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

Table 3-24. PPROTCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	PS27_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
8	PS26_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

Table 3-24. PPROTCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PS25_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
0	PS24_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

3.4.1.13 PCSPWRDWNSET0 Register (Offset = 60h) [Reset = 0000001Fh]

PCSPWRDWNSET0 is shown in Table 3-25.

Return to the Summary Table.

Set-only register to powerdown independent (non-shared) PCS frames 0 to 31

Table 3-25. PCSPWRDWNSET0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

Table 3-25. PCSPWRDWNSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
30	PCS30_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
28	PCS28_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
22	PCS22_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

Table 3-25. PCSPWRDWNSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	PCS20_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
18	PCS18_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
12	PCS12_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

Table 3-25. PCSPWRDWNSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PCS10_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
8	PCS8_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_SET	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_SET	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
2	PCS2_PWRDWN_SET	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_SET	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

Table 3-25. PCSPWRDWNSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PCS0_PWRDWN_SET	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

3.4.1.14 PCSPWRDWNSET1 Register (Offset = 64h) [Reset = 00000000h]

PCSPWRDWNSET1 is shown in Table 3-26.

Return to the Summary Table.

Set-only register to powerdown independent (non-shared) PCS frames 32 to 63

Table 3-26. PCSPWRDWNSET1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

Table 3-26. PCSPWRDWNSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	PCS56_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
22	PCS54_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
21	PCS53_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

Table 3-26. PCSPWRDWNSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	PCS46_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
12	PCS44_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
11	PCS43_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

Table 3-26. PCSPWRDWNSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	PCS36_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
2	PCS34_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
1	PCS33_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

3.4.1.15 PCSPWRDWNCLR0 Register (Offset = 70h) [Reset = 0000001Fh]

PCSPWRDWNCLR0 is shown in Table 3-27.

Return to the Summary Table.

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 0 to 31

Table 3-27. PCSPWRDWNCLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
30	PCS30_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

Table 3-27. PCSPWRDWNCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	PCS28_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
22	PCS22_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
20	PCS20_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

Table 3-27. PCSPWRDWNCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	PCS18_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
12	PCS12_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
10	PCS10_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

Table 3-27. PCSPWRDWNCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PCS8_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_CLR	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_CLR	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
2	PCS2_PWRDWN_CLR	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_CLR	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
0	PCS0_PWRDWN_CLR	R/W	1h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

3.4.1.16 PCSPWRDWNCLR1 Register (Offset = 74h) [Reset = 00000000h]

PCSPWRDWNCLR1 is shown in Table 3-28.

Return to the Summary Table.

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 32 to 63

Table 3-28. PCSPWRDWNCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
24	PCS56_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
22	PCS54_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

Table 3-28. PCSPWRDWNCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	PCS53_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
14	PCS46_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
12	PCS44_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

Table 3-28. PCSPWRDWNCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	PCS43_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
4	PCS36_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
2	PCS34_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

Table 3-28. PCSPWRDWNCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PCS33_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

3.4.1.17 PSPWRDWNSET0 Register (Offset = 80h) [Reset = 00011111h]

PSPWRDWNSET0 is shown in Table 3-29.

Return to the Summary Table.

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS0 to PS7

Table 3-29. PSPWRDWNSET0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

Table 3-29. PSPWRDWNSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	PS6_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
24	PS6_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

Table 3-29. PSPWRDWNSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PS4_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
16	PS4_QUAD0_PWRDWN_SET	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_SET	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

Table 3-29. PSPWRDWNSET0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_SET	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_SET	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
0	PS0_QUAD0_PWRDWN_SET	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

3.4.1.18 PSPWRDWNSET1 Register (Offset = 84h) [Reset = 00000000h]

PSPWRDWNSET1 is shown in Table 3-30.

Return to the Summary Table.

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS8 to PS15

Table 3-30. PSPWRDWNSET1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
24	PS14_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

Table 3-30. PSPWRDWNSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	PS13_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
16	PS12_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

Table 3-30. PSPWRDWNSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	PS11_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
8	PS10_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

Table 3-30. PSPWRDWNSET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PS9_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
0	PS8_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

ADVANCE INFORMATION

3.4.1.19 PSPWRDWNSET2 Register (Offset = 88h) [Reset = 00000000h]

PSPWRDWNSET2 is shown in Table 3-31.

Return to the Summary Table.

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS16 to PS23

Table 3-31. PSPWRDWNSET2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-31. PSPWRDWNSET2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
30	PS23_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
24	PS22_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-31. PSPWRDWNSET2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	PS21_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
16	PS20_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-31. PSPWRDWNSET2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	PS19_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
8	PS18_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-31. PSPWRDWNSET2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PS16_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
0	PS16_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

3.4.1.20 PSPWRDWNSET3 Register (Offset = 8Ch) [Reset = 00000000h]

PSPWRDWNSET3 is shown in Table 3-32.

Return to the Summary Table.

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS24 to PS31

Table 3-32. PSPWRDWNSET3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-32. PSPWRDWNSET3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	PS31_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
24	PS30_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-32. PSPWRDWNSET3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	PS28_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
16	PS28_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-32. PSPWRDWNSET3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PS26_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
8	PS26_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-32. PSPWRDWNSET3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PS24_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
0	PS24_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

3.4.1.21 PSPWRDWNCLR0 Register (Offset = A0h) [Reset = 00011111h]

PSPWRDWNCLR0 is shown in Table 3-33.

Return to the Summary Table.

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS0 to PS7

Table 3-33. PSPWRDWNCLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

Table 3-33. PSPWRDWNCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	PS6_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
24	PS6_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

Table 3-33. PSPWRDWNCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PS4_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
16	PS4_QUAD0_PWRDWN_CLR	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_CLR	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

Table 3-33. PSPWRDWNCLR0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_CLR	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_CLR	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
0	PS0_QUAD0_PWRDWN_CLR	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

3.4.1.22 PSPWRDWNCLR1 Register (Offset = A4h) [Reset = 00000000h]

PSPWRDWNCLR1 is shown in Table 3-34.

Return to the Summary Table.

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS8 to PS15

Table 3-34. PSPWRDWNCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
24	PS14_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

Table 3-34. PSPWRDWNCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	PS13_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
16	PS12_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

Table 3-34. PSPWRDWNCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	PS11_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
8	PS10_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

Table 3-34. PSPWRDWNCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PS9_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
0	PS8_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

3.4.1.23 PSPWRDWNCLR2 Register (Offset = A8h) [Reset = 00000000h]

PSPWRDWNCLR2 is shown in Table 3-35.

Return to the Summary Table.

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS16 to PS23

Table 3-35. PSPWRDWNCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-35. PSPWRDWNCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
30	PS23_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
24	PS22_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-35. PSPWRDWNCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	PS21_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
16	PS20_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-35. PSPWRDWNCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	PS19_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
8	PS18_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

Table 3-35. PSPWRDWNCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PS16_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
0	PS16_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

3.4.1.24 PSPWRDWNCLR3 Register (Offset = ACh) [Reset = 00000000h]

PSPWRDWNCLR3 is shown in Table 3-36.

Return to the Summary Table.

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS24 to PS31

Table 3-36. PSPWRDWNCLR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_CLR	R/W	0h	
30	PS31_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-36. PSPWRDWNCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	PS31_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
24	PS30_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-36. PSPWRDWNCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	PS28_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
16	PS28_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-36. PSPWRDWNCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PS26_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
8	PS26_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

Table 3-36. PSPWRDWNCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PS24_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
0	PS24_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered down. 0 = The clock to the peripheral starting at quadrant ‘m’ of the peripheral frame ‘n’ needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

ADVANCE INFORMATION

3.4.1.25 PDPWRDWNSET Register (Offset = C0h) [Reset = 00000001h]

PDPWRDWNSET is shown in Table 3-37.

Return to the Summary Table.

Set-only register to powerdown the debug frame

Table 3-37. PDPWRDWNSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_SET	R/W	1h	Readable in both user and privileged modes. 1 = Clock to the debug frame needs to be powered down. 0 = Clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get set in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect

3.4.1.26 PDPWRDWNCLR Register (Offset = C4h) [Reset = 00000001h]

PDPWRDWNCLR is shown in Table 3-38.

Return to the Summary Table.

Clear-only register to deassert the debug frame’s powerdown bit

Table 3-38. PDPWRDWNCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_CLR	R/W	1h	Readable in both user and privileged modes. 1 = The clock to the debug frame needs to be powered down. 0 = The clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get cleared in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect

3.4.1.27 MSTIDWRENA Register (Offset = 200h) [Reset = 00000005h]

MSTIDWRENA is shown in Table 3-39.

Return to the Summary Table.

controllerID Protection Write Enable Register

Table 3-39. MSTIDWRENA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTIDREG_WRENA	R/W	5h	Readable in both user and privileged modes. 1010 = All controller-id registers are unlocked and available for write. others = Writes to all controller-id registers are locked. Writable only in privileged mode 1010 = Writes to controller-id registers are unlocked. others = Writes to controller-id registers are locked.

3.4.1.28 MSTIDENA Register (Offset = 204h) [Reset = 00000005h]

MSTIDENA is shown in Table 3-40.

Return to the Summary Table.

controllerID Protection Enable Register

Table 3-40. MSTIDENA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTID_CHK_EN	R/W	5h	Readable in both user and privileged modes. Writable only in privileged mode 1010 = Enable the controller-id feature check. others = controller-id check is disabled.

3.4.1.29 MSTIDDIAGCTRL Register (Offset = 208h) [Reset = 00000005h]

MSTIDDIAGCTRL is shown in Table 3-41.

Return to the Summary Table.

controllerID Diagnostic Control Register

Table 3-41. MSTIDDIAGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-8	DIAG_CMP_VALUE	R/W	0h	controllerID diagnostic mode control register bits; 4-bit data which is compared with the controller-id register of all defined frames during diagnostic mode. Any error in compare logic is indicated through AERROR output from PCR. Readable in both user and privileged modes. Reads the programmed value in diagnostic compare value field. Writable only in privileged mode
7-4	RESERVED	R	0h	Reserved
3-0	DIAG_MODE_EN	R/W	5h	controllerID compare logic diagnostic mode enable bits; 4-bit key for enabling the controller-id registers compare logic. Readable in both user and privileged modes. Writable only in privileged mode 1010 = controller-id compare diagnostic mode is enabled. others = controller-id compare diagnostic mode is disabled.

3.4.1.30 PS0MSTID_L Register (Offset = 300h) [Reset = 0000FFFFh]

PS0MSTID_L is shown in Table 3-42.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register0_L

Table 3-42. PS0MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS0_QUAD0_MSTID	R/W	FFFFh	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.31 PS0MSTID_H Register (Offset = 304h) [Reset = 00000000h]

PS0MSTID_H is shown in Table 3-43.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register0_H

Table 3-43. PS0MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-43. PS0MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.32 PS1MSTID_L Register (Offset = 308h) [Reset = 0000FFFFh]

PS1MSTID_L is shown in Table 3-44.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register1_L

Table 3-44. PS1MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD0_MSTID	R/W	FFFFh	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.33 PS1MSTID_H Register (Offset = 30Ch) [Reset = 00000000h]

PS1MSTID_H is shown in Table 3-45.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register1_H

Table 3-45. PS1MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.34 PS2MSTID_L Register (Offset = 310h) [Reset = 0000FFFFh]

PS2MSTID_L is shown in Table 3-46.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register2_L

Table 3-46. PS2MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-46. PS2MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS2_QUAD0_MSTID	R/W	FFFFh	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.35 PS2MSTID_H Register (Offset = 314h) [Reset = 00000000h]

PS2MSTID_H is shown in Table 3-47.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register2_H

Table 3-47. PS2MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.36 PS3MSTID_L Register (Offset = 318h) [Reset = 0000FFFFh]

PS3MSTID_L is shown in Table 3-48.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register3_L

Table 3-48. PS3MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS3_QUAD0_MSTID	R/W	FFFFh	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.37 PS3MSTID_H Register (Offset = 31Ch) [Reset = 00000000h]

PS3MSTID_H is shown in Table 3-49.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register3_H

Table 3-49. PS3MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-49. PS3MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.38 PS4MSTID_L Register (Offset = 320h) [Reset = 0000FFFFh]

PS4MSTID_L is shown in Table 3-50.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register4_L

Table 3-50. PS4MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD0_MSTID	R/W	FFFFh	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.39 PS4MSTID_H Register (Offset = 324h) [Reset = 00000000h]

PS4MSTID_H is shown in Table 3-51.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register4_H

Table 3-51. PS4MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.40 PS5MSTID_L Register (Offset = 328h) [Reset = 00000000h]

PS5MSTID_L is shown in Table 3-52.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register5_L

Table 3-52. PS5MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-52. PS5MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.41 PS5MSTID_H Register (Offset = 32Ch) [Reset = 00000000h]

PS5MSTID_H is shown in Table 3-53.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register5_H

Table 3-53. PS5MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.42 PS6MSTID_L Register (Offset = 330h) [Reset = 00000000h]

PS6MSTID_L is shown in Table 3-54.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register6_L

Table 3-54. PS6MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.43 PS6MSTID_H Register (Offset = 334h) [Reset = 00000000h]

PS6MSTID_H is shown in Table 3-55.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register6_H

Table 3-55. PS6MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-55. PS6MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.44 PS7MSTID_L Register (Offset = 338h) [Reset = 00000000h]

PS7MSTID_L is shown in Table 3-56.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register7_L

Table 3-56. PS7MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.45 PS7MSTID_H Register (Offset = 33Ch) [Reset = 00000000h]

PS7MSTID_H is shown in Table 3-57.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register7_H

Table 3-57. PS7MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.46 PS8MSTID_L Register (Offset = 340h) [Reset = 00000000h]

PS8MSTID_L is shown in Table 3-58.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register8_L

Table 3-58. PS8MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-58. PS8MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS8_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.47 PS8MSTID_H Register (Offset = 344h) [Reset = 00000000h]

PS8MSTID_H is shown in Table 3-59.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register8_H

Table 3-59. PS8MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS8_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.48 PS9MSTID_L Register (Offset = 348h) [Reset = 00000000h]

PS9MSTID_L is shown in Table 3-60.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register9_L

Table 3-60. PS9MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS9_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.49 PS9MSTID_H Register (Offset = 34Ch) [Reset = 00000000h]

PS9MSTID_H is shown in Table 3-61.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register9_H

Table 3-61. PS9MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-61. PS9MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS9_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.50 PS10MSTID_L Register (Offset = 350h) [Reset = 00000000h]

PS10MSTID_L is shown in Table 3-62.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register10_L

Table 3-62. PS10MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.51 PS10MSTID_H Register (Offset = 354h) [Reset = 00000000h]

PS10MSTID_H is shown in Table 3-63.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register10_H

Table 3-63. PS10MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.52 PS11MSTID_L Register (Offset = 358h) [Reset = 00000000h]

PS11MSTID_L is shown in Table 3-64.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register11_L

Table 3-64. PS11MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-64. PS11MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS11_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.53 PS11MSTID_H Register (Offset = 35Ch) [Reset = 00000000h]

PS11MSTID_H is shown in Table 3-65.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register11_H

Table 3-65. PS11MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS11_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.54 PS12MSTID_L Register (Offset = 360h) [Reset = 00000000h]

PS12MSTID_L is shown in Table 3-66.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register12_L

Table 3-66. PS12MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS12_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.55 PS12MSTID_H Register (Offset = 364h) [Reset = 00000000h]

PS12MSTID_H is shown in Table 3-67.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register12_H

Table 3-67. PS12MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-67. PS12MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS12_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.56 PS13MSTID_L Register (Offset = 368h) [Reset = 00000000h]

PS13MSTID_L is shown in Table 3-68.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register13_L

Table 3-68. PS13MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.57 PS13MSTID_H Register (Offset = 36Ch) [Reset = 00000000h]

PS13MSTID_H is shown in Table 3-69.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register13_H

Table 3-69. PS13MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.58 PS14MSTID_L Register (Offset = 370h) [Reset = 00000000h]

PS14MSTID_L is shown in Table 3-70.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register14_L

Table 3-70. PS14MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-70. PS14MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS14_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.59 PS14MSTID_H Register (Offset = 374h) [Reset = 00000000h]

PS14MSTID_H is shown in Table 3-71.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register14_H

Table 3-71. PS14MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS14_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.60 PS15MSTID_L Register (Offset = 378h) [Reset = 00000000h]

PS15MSTID_L is shown in Table 3-72.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register15_L

Table 3-72. PS15MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS15_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.61 PS15MSTID_H Register (Offset = 37Ch) [Reset = 00000000h]

PS15MSTID_H is shown in Table 3-73.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register15_H

Table 3-73. PS15MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-73. PS15MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS15_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.62 PS16MSTID_L Register (Offset = 380h) [Reset = 00000000h]

PS16MSTID_L is shown in Table 3-74.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register16_L

Table 3-74. PS16MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.63 PS16MSTID_H Register (Offset = 384h) [Reset = 00000000h]

PS16MSTID_H is shown in Table 3-75.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register16_H

Table 3-75. PS16MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.64 PS17MSTID_L Register (Offset = 388h) [Reset = 00000000h]

PS17MSTID_L is shown in Table 3-76.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register17_L

Table 3-76. PS17MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-76. PS17MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS17_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.65 PS17MSTID_H Register (Offset = 38Ch) [Reset = 00000000h]

PS17MSTID_H is shown in Table 3-77.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register17_H

Table 3-77. PS17MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS17_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.66 PS18MSTID_L Register (Offset = 390h) [Reset = 00000000h]

PS18MSTID_L is shown in Table 3-78.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register18_L

Table 3-78. PS18MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS18_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.67 PS18MSTID_H Register (Offset = 394h) [Reset = 00000000h]

PS18MSTID_H is shown in Table 3-79.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register18_H

Table 3-79. PS18MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-79. PS18MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS18_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.68 PS19MSTID_L Register (Offset = 398h) [Reset = 00000000h]

PS19MSTID_L is shown in Table 3-80.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register19_L

Table 3-80. PS19MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.69 PS19MSTID_H Register (Offset = 39Ch) [Reset = 00000000h]

PS19MSTID_H is shown in Table 3-81.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register19_H

Table 3-81. PS19MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.70 PS20MSTID_L Register (Offset = 3A0h) [Reset = 00000000h]

PS20MSTID_L is shown in Table 3-82.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register20_L

Table 3-82. PS20MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-82. PS20MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS20_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.71 PS20MSTID_H Register (Offset = 3A4h) [Reset = 00000000h]

PS20MSTID_H is shown in Table 3-83.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register20_H

Table 3-83. PS20MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS20_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.72 PS21MSTID_L Register (Offset = 3A8h) [Reset = 00000000h]

PS21MSTID_L is shown in Table 3-84.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register21_L

Table 3-84. PS21MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS21_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.73 PS21MSTID_H Register (Offset = 3ACh) [Reset = 00000000h]

PS21MSTID_H is shown in Table 3-85.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register21_H

Table 3-85. PS21MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-85. PS21MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS21_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.74 PS22MSTID_L Register (Offset = 3B0h) [Reset = 00000000h]

PS22MSTID_L is shown in Table 3-86.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register22_L

Table 3-86. PS22MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.75 PS22MSTID_H Register (Offset = 3B4h) [Reset = 00000000h]

PS22MSTID_H is shown in Table 3-87.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register22_H

Table 3-87. PS22MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.76 PS23MSTID_L Register (Offset = 3B8h) [Reset = 00000000h]

PS23MSTID_L is shown in Table 3-88.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register23_L

Table 3-88. PS23MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-88. PS23MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS23_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.77 PS23MSTID_H Register (Offset = 3BCh) [Reset = 00000000h]

PS23MSTID_H is shown in Table 3-89.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register23_H

Table 3-89. PS23MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS23_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.78 PS24MSTID_L Register (Offset = 3C0h) [Reset = 00000000h]

PS24MSTID_L is shown in Table 3-90.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register24_L

Table 3-90. PS24MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS24_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.79 PS24MSTID_H Register (Offset = 3C4h) [Reset = 00000000h]

PS24MSTID_H is shown in Table 3-91.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register24_H

Table 3-91. PS24MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-91. PS24MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS24_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.80 PS25MSTID_L Register (Offset = 3C8h) [Reset = 00000000h]

PS25MSTID_L is shown in Table 3-92.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register25_L

Table 3-92. PS25MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.81 PS25MSTID_H Register (Offset = 3CCh) [Reset = 00000000h]

PS25MSTID_H is shown in Table 3-93.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register25_H

Table 3-93. PS25MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.82 PS26MSTID_L Register (Offset = 3D0h) [Reset = 00000000h]

PS26MSTID_L is shown in Table 3-94.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register26_L

Table 3-94. PS26MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-94. PS26MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS26_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.83 PS26MSTID_H Register (Offset = 3D4h) [Reset = 00000000h]

PS26MSTID_H is shown in Table 3-95.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register26_H

Table 3-95. PS26MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS26_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.84 PS27MSTID_L Register (Offset = 3D8h) [Reset = 00000000h]

PS27MSTID_L is shown in Table 3-96.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register27_L

Table 3-96. PS27MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS27_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.85 PS27MSTID_H Register (Offset = 3DCh) [Reset = 00000000h]

PS27MSTID_H is shown in Table 3-97.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register27_H

Table 3-97. PS27MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-97. PS27MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS27_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.86 PS28MSTID_L Register (Offset = 3E0h) [Reset = 00000000h]

PS28MSTID_L is shown in Table 3-98.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register28_L

Table 3-98. PS28MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.87 PS28MSTID_H Register (Offset = 3E4h) [Reset = 00000000h]

PS28MSTID_H is shown in Table 3-99.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register28_H

Table 3-99. PS28MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.88 PS29MSTID_L Register (Offset = 3E8h) [Reset = 00000000h]

PS29MSTID_L is shown in Table 3-100.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register29_L

Table 3-100. PS29MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-100. PS29MSTID_L Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS29_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.89 PS29MSTID_H Register (Offset = 3ECh) [Reset = 00000000h]

PS29MSTID_H is shown in Table 3-101.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register29_H

Table 3-101. PS29MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS29_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.90 PS30MSTID_L Register (Offset = 3F0h) [Reset = 0000FFFFh]

PS30MSTID_L is shown in Table 3-102.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register30_L

Table 3-102. PS30MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS30_QUAD0_MSTID	R/W	FFFFh	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.91 PS30MSTID_H Register (Offset = 3F4h) [Reset = 00000000h]

PS30MSTID_H is shown in Table 3-103.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register30_H

Table 3-103. PS30MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

Table 3-103. PS30MSTID_H Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PS30_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.92 PS31MSTID_L Register (Offset = 3F8h) [Reset = 00000000h]

PS31MSTID_L is shown in Table 3-104.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register31_L

Table 3-104. PS31MSTID_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.93 PS31MSTID_H Register (Offset = 3FCh) [Reset = 00000000h]

PS31MSTID_H is shown in Table 3-105.

Return to the Summary Table.

Peripheral Frame controller-ID Protection Register31_H

Table 3-105. PS31MSTID_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 controllers to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by controllers with matching controller-ID. 0 = The peipheral is locked for controllers with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

3.4.1.94 PCS0MSTID Register (Offset = 540h) [Reset = FFFFFFFFh]

PCS0MSTID is shown in Table 3-106.

Return to the Summary Table.

Memory Frame controller ID Protection Register0

Table 3-106. PCS0MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS1MSTID	R/W	FFFFh	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-106. PCS0MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS0MSTID	R/W	FFFFh	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.95 PCS1MSTID Register (Offset = 544h) [Reset = FFFFFFFFh]

PCS1MSTID is shown in Table 3-107.

Return to the Summary Table.

Memory Frame controller ID Protection Register1

Table 3-107. PCS1MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS3MSTID	R/W	FFFFh	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS2MSTID	R/W	FFFFh	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.96 PCS2MSTID Register (Offset = 548h) [Reset = 0000FFFFh]

PCS2MSTID is shown in Table 3-108.

Return to the Summary Table.

Memory Frame controller ID Protection Register2

Table 3-108. PCS2MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS5MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS4MSTID	R/W	FFFFh	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.97 PCS3MSTID Register (Offset = 54Ch) [Reset = 00000000h]

PCS3MSTID is shown in Table 3-109.

Return to the Summary Table.

Memory Frame controller ID Protection Register3

Table 3-109. PCS3MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS7MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-109. PCS3MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS6MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.98 PCS4MSTID Register (Offset = 550h) [Reset = 00000000h]

PCS4MSTID is shown in Table 3-110.

Return to the Summary Table.

Memory Frame controller ID Protection Register4

Table 3-110. PCS4MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS9MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS8MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.99 PCS5MSTID Register (Offset = 554h) [Reset = 00000000h]

PCS5MSTID is shown in Table 3-111.

Return to the Summary Table.

Memory Frame controller ID Protection Register5

Table 3-111. PCS5MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS11MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS10MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.100 PCS6MSTID Register (Offset = 558h) [Reset = 00000000h]

PCS6MSTID is shown in Table 3-112.

Return to the Summary Table.

Memory Frame controller ID Protection Register6

Table 3-112. PCS6MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS13MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-112. PCS6MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS12MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.101 PCS7MSTID Register (Offset = 55Ch) [Reset = 00000000h]

PCS7MSTID is shown in Table 3-113.

Return to the Summary Table.

Memory Frame controller ID Protection Register7

Table 3-113. PCS7MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS15MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS14MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.102 PCS8MSTID Register (Offset = 560h) [Reset = 00000000h]

PCS8MSTID is shown in Table 3-114.

Return to the Summary Table.

Memory Frame controller ID Protection Register8

Table 3-114. PCS8MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS17MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS16MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.103 PCS9MSTID Register (Offset = 564h) [Reset = 00000000h]

PCS9MSTID is shown in Table 3-115.

Return to the Summary Table.

Memory Frame controller ID Protection Register9

Table 3-115. PCS9MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS19MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-115. PCS9MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS18MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.104 PCS10MSTID Register (Offset = 568h) [Reset = 00000000h]

PCS10MSTID is shown in Table 3-116.

Return to the Summary Table.

Memory Frame controller ID Protection Register10

Table 3-116. PCS10MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS21MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS20MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.105 PCS11MSTID Register (Offset = 56Ch) [Reset = 00000000h]

PCS11MSTID is shown in Table 3-117.

Return to the Summary Table.

Memory Frame controller ID Protection Register11

Table 3-117. PCS11MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS23MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS22MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.106 PCS12MSTID Register (Offset = 570h) [Reset = 00000000h]

PCS12MSTID is shown in Table 3-118.

Return to the Summary Table.

Memory Frame controller ID Protection Register12

Table 3-118. PCS12MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS25MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-118. PCS12MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS24MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.107 PCS13MSTID Register (Offset = 574h) [Reset = 00000000h]

PCS13MSTID is shown in Table 3-119.

Return to the Summary Table.

Memory Frame controller ID Protection Register13

Table 3-119. PCS13MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS27MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS26MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.108 PCS14MSTID Register (Offset = 578h) [Reset = 00000000h]

PCS14MSTID is shown in Table 3-120.

Return to the Summary Table.

Memory Frame controller ID Protection Register14

Table 3-120. PCS14MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS29MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS28MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.109 PCS15MSTID Register (Offset = 57Ch) [Reset = 00000000h]

PCS15MSTID is shown in Table 3-121.

Return to the Summary Table.

Memory Frame controller ID Protection Register15

Table 3-121. PCS15MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS31MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-121. PCS15MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS30MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.110 PCS16MSTID Register (Offset = 580h) [Reset = 00000000h]

PCS16MSTID is shown in Table 3-122.

Return to the Summary Table.

Memory Frame controller ID Protection Register16

Table 3-122. PCS16MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS33MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS32MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.111 PCS17MSTID Register (Offset = 584h) [Reset = 00000000h]

PCS17MSTID is shown in Table 3-123.

Return to the Summary Table.

Memory Frame controller ID Protection Register17

Table 3-123. PCS17MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS35MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS34MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.112 PCS18MSTID Register (Offset = 588h) [Reset = 00000000h]

PCS18MSTID is shown in Table 3-124.

Return to the Summary Table.

Memory Frame controller ID Protection Register18

Table 3-124. PCS18MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS37MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-124. PCS18MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS36MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.113 PCS19MSTID Register (Offset = 58Ch) [Reset = 00000000h]

PCS19MSTID is shown in Table 3-125.

Return to the Summary Table.

Memory Frame controller ID Protection Register19

Table 3-125. PCS19MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS39MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS38MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.114 PCS20MSTID Register (Offset = 590h) [Reset = 00000000h]

PCS20MSTID is shown in Table 3-126.

Return to the Summary Table.

Memory Frame controller ID Protection Register20

Table 3-126. PCS20MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS41MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS40MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.115 PCS21MSTID Register (Offset = 594h) [Reset = 00000000h]

PCS21MSTID is shown in Table 3-127.

Return to the Summary Table.

Memory Frame controller ID Protection Register21

Table 3-127. PCS21MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS43MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-127. PCS21MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS42MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.116 PCS22MSTID Register (Offset = 598h) [Reset = 00000000h]

PCS22MSTID is shown in Table 3-128.

Return to the Summary Table.

Memory Frame controller ID Protection Register22

Table 3-128. PCS22MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS45MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS44MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.117 PCS23MSTID Register (Offset = 59Ch) [Reset = 00000000h]

PCS23MSTID is shown in Table 3-129.

Return to the Summary Table.

Memory Frame controller ID Protection Register23

Table 3-129. PCS23MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS47MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS46MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.118 PCS24MSTID Register (Offset = 5A0h) [Reset = 00000000h]

PCS24MSTID is shown in Table 3-130.

Return to the Summary Table.

Memory Frame controller ID Protection Register24

Table 3-130. PCS24MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS49MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-130. PCS24MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS48MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.119 PCS25MSTID Register (Offset = 5A4h) [Reset = 00000000h]

PCS25MSTID is shown in Table 3-131.

Return to the Summary Table.

Memory Frame controller ID Protection Register25

Table 3-131. PCS25MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS51MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS50MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.120 PCS26MSTID Register (Offset = 5A8h) [Reset = 00000000h]

PCS26MSTID is shown in Table 3-132.

Return to the Summary Table.

Memory Frame controller ID Protection Register26

Table 3-132. PCS26MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS53MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS52MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.121 PCS27MSTID Register (Offset = 5ACh) [Reset = 00000000h]

PCS27MSTID is shown in Table 3-133.

Return to the Summary Table.

Memory Frame controller ID Protection Register27

Table 3-133. PCS27MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS55MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-133. PCS27MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS54MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.122 PCS28MSTID Register (Offset = 5B0h) [Reset = 00000000h]

PCS28MSTID is shown in Table 3-134.

Return to the Summary Table.

Memory Frame controller ID Protection Register28

Table 3-134. PCS28MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS57MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS56MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.123 PCS29MSTID Register (Offset = 5B4h) [Reset = 00000000h]

PCS29MSTID is shown in Table 3-135.

Return to the Summary Table.

Memory Frame controller ID Protection Register29

Table 3-135. PCS29MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS59MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS58MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.124 PCS30MSTID Register (Offset = 5B8h) [Reset = 00000000h]

PCS30MSTID is shown in Table 3-136.

Return to the Summary Table.

Memory Frame controller ID Protection Register30

Table 3-136. PCS30MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS61MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

Table 3-136. PCS30MSTID Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	PCS60MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.125 PCS31MSTID Register (Offset = 5BCh) [Reset = 00000000h]

PCS31MSTID is shown in Table 3-137.

Return to the Summary Table.

Memory Frame controller ID Protection Register31

Table 3-137. PCS31MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCS63MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS62MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 controllers to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by controllers with controller-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by controllers with controller-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

3.4.1.126 PCREXTMSTID Register (Offset = 5E0h) [Reset = 00000000h]

PCREXTMSTID is shown in Table 3-138.

Return to the Summary Table.

controller-ID Protection Register for external PCR

Table 3-138. PCREXTMSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PCREXT_MSTID	R/W	0h	These bits sets the permission for maximum of 16 controllers to address the external PCR frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by controller with matching controller-ID. 0 = The memory is locked for controller with matching controller-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

4 Device Initialization

4.1 Initialization Overview

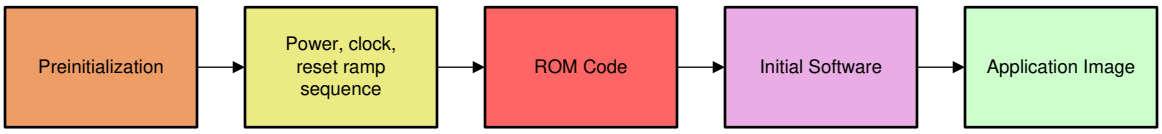


Figure 4-1. Device Initialization

Below is an overview of the initialization process and its steps:

- **Preinitialization:** Power, clock, and control connections must be present, and the boot configuration pins must be held at the desired logical levels.
- **Power, clock, reset ramp sequence:** Specific sequence that is applied by the power-management chip
- **ROM code:** Responsible for finding, downloading, and executing the initial software (RBL)
- **Application Image:** The application that runs on the main core/processor.

The first two steps in the initialization process are hardware-oriented; however, they require an understanding of the process of configuring these system interface pins (balls on the device), which have software-configurable functionality. This configuration is an essential part of the chip configuration and is application-dependent. This chapter discusses these system-interface pins, the associated configuration registers, and memory structures that are vital to the correct initialization of the device.

4.2 ROM Code Overview

ROM bootloader (or RBL) is a software that resides in a on-chip read-only memory (ROM) of application subsystem (APPSS) to assist the customer in transferring and executing their application code. Key features supported by RBL are the following –

- Downloading customer application consisting of binary images for different sub-systems from serial flash.
- Downloading customer application consisting of binary images for different sub-systems over SPI or UART interfaces if serial flash is not detected.
- Support for serial flash programming in device management mode of operation over UART interface.
- Wakeup support from deep sleep mode.
- Handling of warm and soft reset scenarios.
- Support for append mode of operations.
- Support for safety features such as self-test of memories and CPU cores, ECC protection for memories, fault handling etc.

To accommodate various system scenarios, the ROM code supports several boot modes. These boot modes can be broadly classified as:

- Autonomous boot modes
- Secondary boot modes

In case of autonomous mode, the device boots from serial flash over the QSPI interface. In case of secondary mode, the device boots either over SPI or UART interface connected to the external host MCU.

In all boot modes, the entire boot operation can be partitioned into two sections:

- Hardware initialization phase
- Boot process

During initialization, the ROM code configures the device resources (PLLs, peripherals, pins) as needed to support the boot process. The resources used depend on the boot mode requirements. During the boot process, the boot image can be loaded into device memory and executed, depending on the boot peripheral.

The main configuration source for boot after power-up are the SOP mode pins which are sampled automatically after reset release and stored in device status registers. At ROM code startup, these pin values are read from the registers to create the boot peripheral list, and the boot configuration tables used later to initialize and startup the PLLs and boot peripherals.

4.3 Boot Modes

The device supports 2 boot modes, Device Management Mode and Application 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 below for the SOP pin values for each mode.

SOP Mode		Bootloader Mode and Description
PMIC_CLK_OUT(default '0')	TDO (default '1')	
0	0	Device Management Mode: This mode is similar to Application mode but ROM code knows that this mode would be used for device management and hence configures the device accordingly (pin muxing, clocks, IP enables etc) and starts expecting command over the UART2 interface.
0	1	Application Mode: Device would power up in functional mode. PORG and PRCM would perform normal operation identifying the Power Management topology and Reference clock system topologies. Once CPU is up, device goes through the device boot sequence downloading the image from external flash/Host Interface(SPI/UART). Download and boot RBL from QSPI flash. Attempt Primary RBL, followed by Secondary RBL if primary loading fails: If above is not successful. Once the boot is completed, device goes into the appropriate mode (Deep sleep or Idle mode) as directed by the loaded application.

4.4 High-Level Bootloader Flow

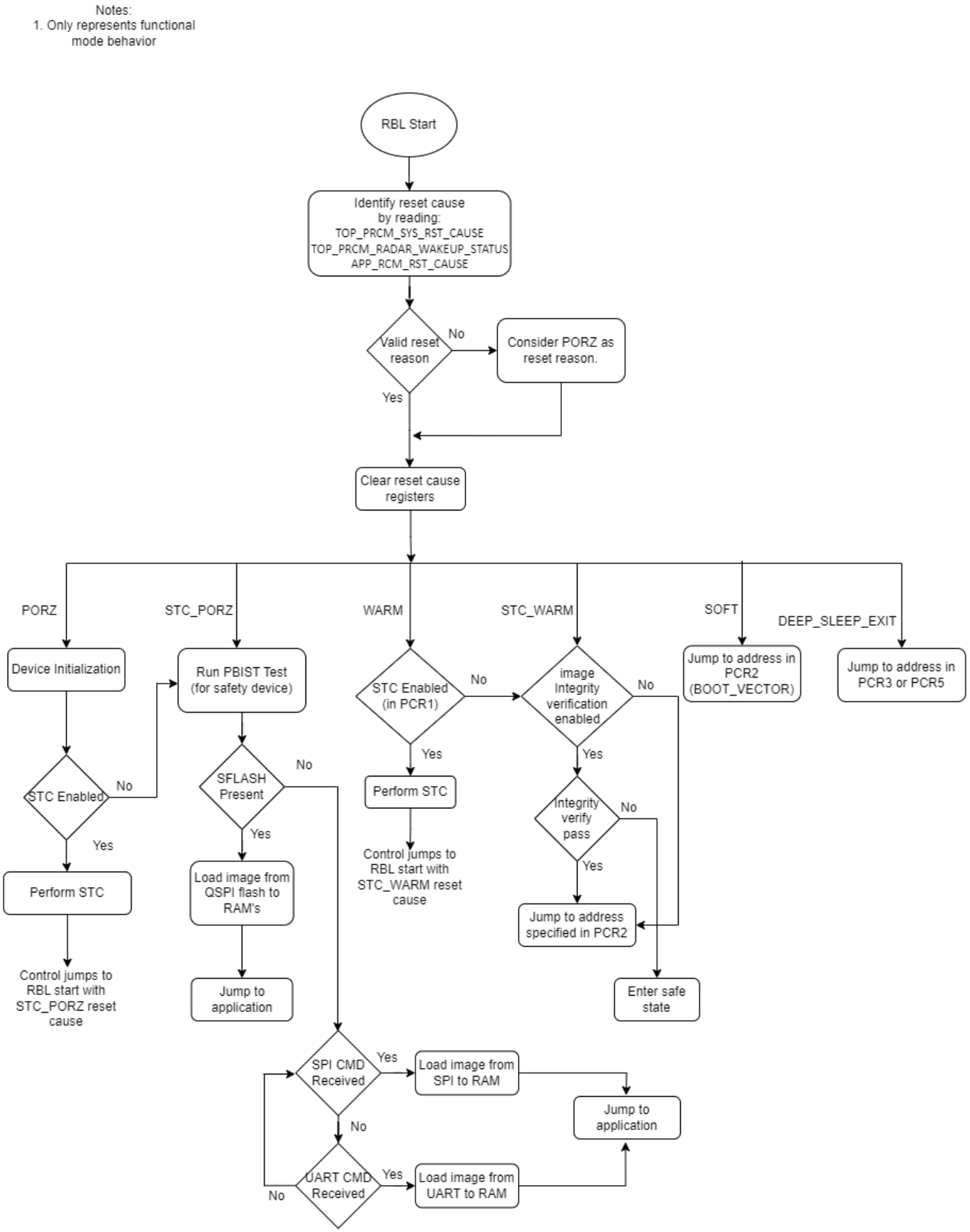


Figure 4-2. Bootloader Flow xWRLx432

Figure 4-2 shows the functional mode behavior of the RBL. This flow is not changeable by the user, as it is flashed into ROM of the device.

Integrity Verification

The RBL makes the following assumptions when performing the integrity check in the STC_WARM reset flow:

- RBL reads the boot vector from TOP_PRCM:PC_REGISTER2[24:0] to determine the boot vector to jump to.
- RBL will perform the integrity check for the APPSS image only
- RBL computes the CRC of the APPSS image based on image length (starting from boot vector) field in TOP_PRCM:PC_REGISTER3[31:0] and compares it with the data integrity check field TOP_PRCM:PC_REGISTER4[31:0]. It is expected that the application populates this register.
- The parity of TOP_PRCM:PC_REGISTER3 and TOP_PRCM:PC_REGISTER4 is expected to be correctly populated in TOP_PRCM:PC_REGISTER1[23:20] and TOP_PRCM:PC_REGISTER2[19:16] by the application.

See Figure 4-9 for more information on the PC_REGISTER fields.

Warm Reset Behavior

On a warm reset, the RBL will perform PBIST checks (if necessary) and verify the application image (if check is enabled). Execution then starts at the beginning of the address pointed to by the boot vector (typically, the start of application image). In this flow, there is necessary HW infrastructure provisioned to ensure that the memory contents are retained on a warm reset cycle with the power to the device maintained to be intact, eliminating the need to a reload the entire application image from Flash.

Note

In the above RBL flow, the image is not reloaded from Flash memory on a warm reset. The RBL will only load from Flash if it reads a POR as the reset reason or if reset reason is invalid.

4.4.1 Application Load from Flash on Warm Reset

A software-based workaround to load the image from Flash is possible by modifying the jump address used in the warm reset flow. Proper care must also be taken to ensure that the FECSS (Front End Controller Subsystem) is in the correct state. A user must perform any necessary safety assessment of their own system to determine if this software-based workaround will meet their needs. The high level flow of the workaround sequence is as follows, and would be performed at the start of application code, if a warm reset has occured:

1. Gracefully power down FEC and HWA
2. Power on FECSS
3. Write Boot vector to TOP_PRCM:PC_REGISTER2[24:0] = 0
4. Write boot vector parity to TOP_PRCM:PC_REGISTER1[15:12] = 0
5. Disable image integrity check - TOP_PRCM:PC_REGISTER1[5:3] = 0
6. Disable STC test - TOP_PRCM:PC_REGISTER1[8:6] = 0
7. Trigger warm reset - TOP_PRCM:RST_SOFT_RESET[0] = 1

After step 7 above, the execution re-enters the RBL, which will start execution as shown in Figure 4-2. The RBL determines the reset reason as invalid because the reset registers have been clear. It will assume a POR, and the application image is loaded from Flash once again. In this flow, two separate warm resets occur.

4.5 Device Management / Flashing SOP Mode (SOP00, Serial Flash Programming)

User application components (loaded to the M4F) are expected to be stored in the serial data flash (SDF) interfaced to the xWRLx432 device over the quad serial peripheral interface (QSPI) interface.

4.5.1 UART Communication Protocol

The UART communication protocol involves a simple command-response flow. The host first sends the command packet to the xWRLx432 device and then waits for a response from the device. Each command packet is immediately processed by the bootloader without any scheduling or processing in the background and

then a valid response packet is sent. The response can either be an ACK, a NACK or it may contain some specific response information. Each command has a known response packet and those are documented in Section 4.5.2 and Section 4.6.3.2.

4.5.2 Bootloader Commands

There are two main packet formats for the UART bootloader, command packets and response packets. Command packets follow the structure below:

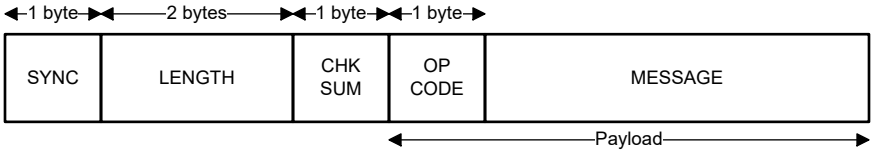


Figure 4-3. Command Packet Format

The command packet consists of a sync pattern (1 byte), the length of the packet (2 bytes), checksum for the payload bytes (1 byte) and the payload. The max size of the payload is 252 bytes. The 2 bytes containing length information are transmitted in big endian format with MSB byte transmitted followed by the LSB byte.

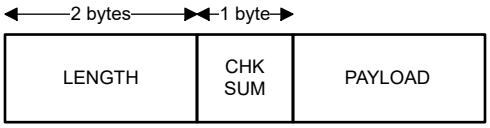
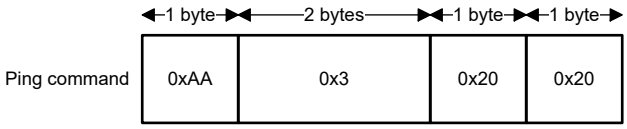
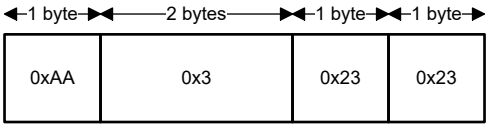
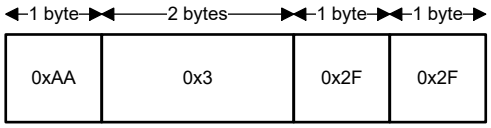
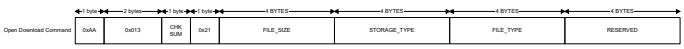
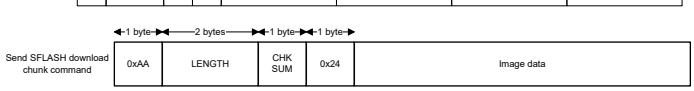
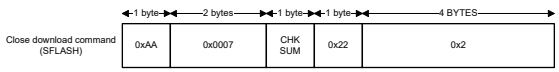


Figure 4-4. Response Packet Format

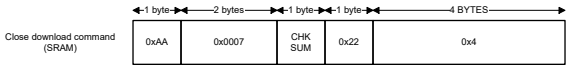
The response packet consists of the length of the packet, checksum for the payload bytes and a payload. For ACK and NACK the payload is 2 bytes.

Supported Command

See more details on these commands and the expected responses for these commands in Section 4.6.3.2. Multi-byte fields are transmitted in big endian format with MSB bytes transmitted followed by the LSB bytes.

Packet	Structure
Ping	
Get Status	
Get Version	
Open Download	
Send SFLASH Download Chunk	
Send SRAM Download Chunk	
Close Download (SFLASH)	

ADVANCE INFORMATION

Packet	Structure
Close Download (SRAM)	

4.5.2.1 Example Command Sequence

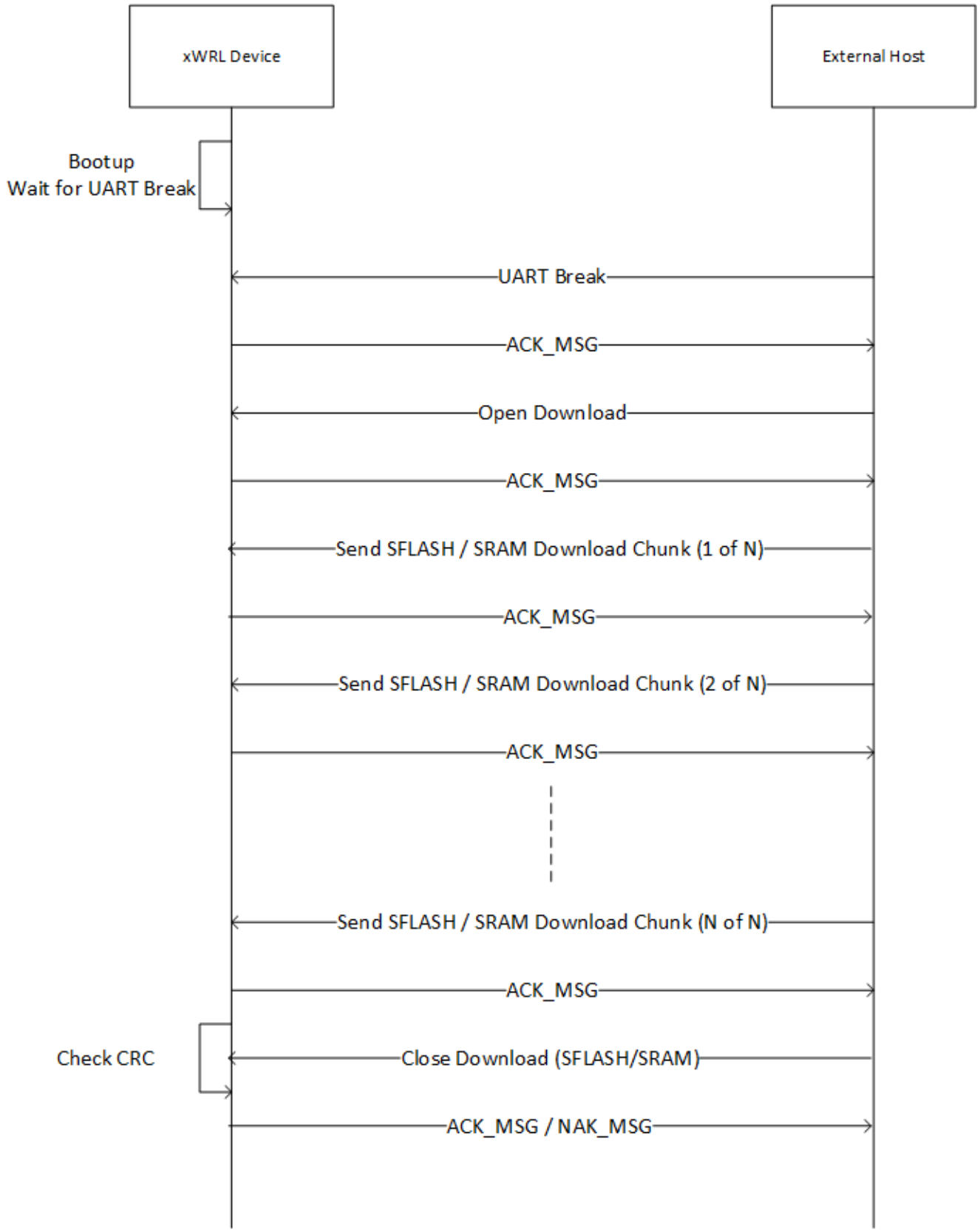


Figure 4-5. UART Download Sequence

4.6 Application/Functional SOP1 Mode

4.6.1 Power On Reset (Booting Application Image)

4.6.1.1 Booting from Serial Flash

Image Load Sequence

In functional mode, the bootloading of an image from the SDF is the first bootmode attempted by the bootloader (see Figure 4-6).

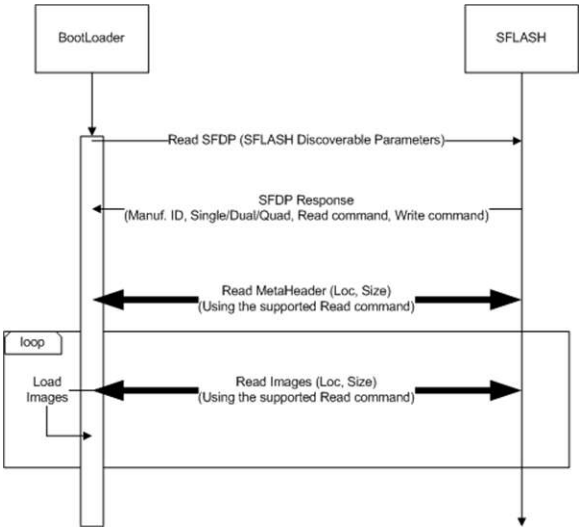


Figure 4-6. Image Load Sequence

This bootmode involves the following steps:

1. Pinmux the QSPI pins of the xWRLx432 device:
2. QSPI is set up to operate at (system clock / 2) = (160/2) = 80 MHz.
3. The SFLASH discoverable parameters (SFDP) command is issued to retrieve the JEDEC compliant response, which includes information regarding the SFLASH capabilities and command set. When the SFDP response is received, the information is used to communicate with the SDF and further interpret the contents and load the images.

Key points

- The ROM bootloader performs the read from the SDF, based on the highest capability mode (quad, dual, or single) as published by the SDF in response to the SFDP command.
- For SDF variants that support quad mode, the quad mode commands are issued; if the quad enable (QE) bit is not set, the communication will fail. In such cases, the load flow assumes that the QE bit in the SDF is already set.
- Fallback images: the bootloader supports loading of images from the following locations as a fallback mechanism if one of the images is corrupted in the SDF. The locations of the images are:
  - META IMG1(SDF offset – 0x0)
  - META IMG2(SDF offset – 0x80000)
  - META IMG3(SDF offset – 0x100000)
  - META IMG4(SDF offset – 0x180000)

See Boot Image Format for image format details.

4.6.1.2 ROM-Assisted Image Download Sequence

The ROM-assisted image download sequence is entered by placing the device in flashing mode. See Programming Serial Data Flash Over UART (Bootloader Service), for further details on the handshake with an external host to receive the image. Figure 4-7 shows the communication with the SDF.

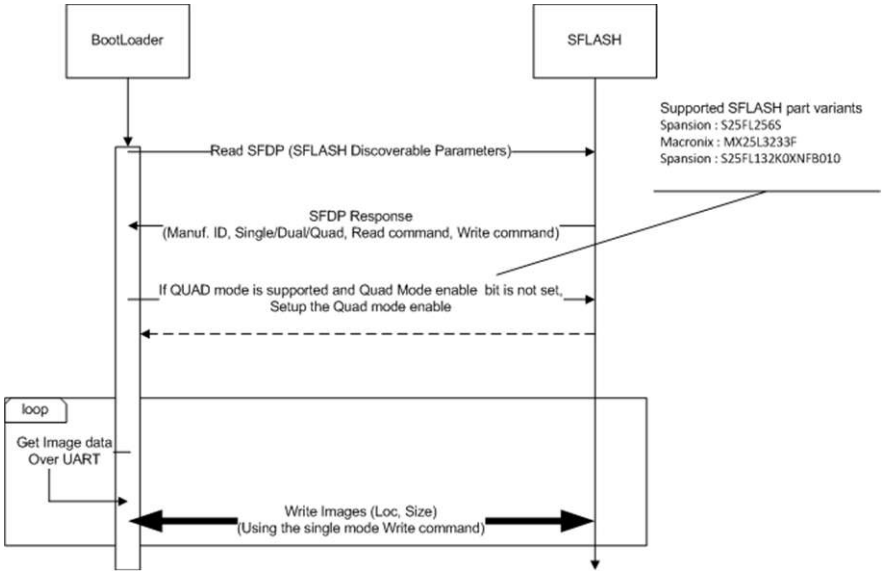


Figure 4-7. ROM-Assisted Image Download Sequence

Key points

- The ROM-assisted download should work with all flash variants that allow for *memory-mapped mode* and *Page program command (0x2)*, with one dummy byte and 24-bit addressing.
- Setting the QE bit varies from one SDF vendor to another. The ROM bootloader supports setting the QE bit for Spansion® and Macronix® variants (certain specific part variants only) in this flow.
- In addition to a checksum-based integrity check for every packet received over the UART, a CRC32-based integrity check is performed over the complete image. The CRC32 is computed incrementally as the packets are received and written to the SDF.

4.6.2 Booting Over SPI

When not using a QSPI flash device to store the application image, the bootloader will look for an image to be loaded via SPI or UART interface, if the first ping/command is initiated through SPI interface then RBL switch to SPI mode of image download. This image can be loaded in from a host PC or processor using the protocol documented below. A simplified diagram for command response flow is shown below.

4.6.2.1 SPI Communication Protocol

The bootloader implements the standard SPI protocol as detailed in Multichannel Serial Port Interface (McSPI). The specific settings used for the SPI interface to the bootloader are below.

Setting	Value
SPI Mode	0 (Polarity 0, Phase 0)
SPI Clock Frequency	5 MHz - 22 MHz (Normal Image) 5 MHz - 11 MHz (Authenticated Image)

4.6.2.2 Example Host Application Flow

An example host command flow for loading an image over SPI can be found below.

ADVANCE INFORMATION

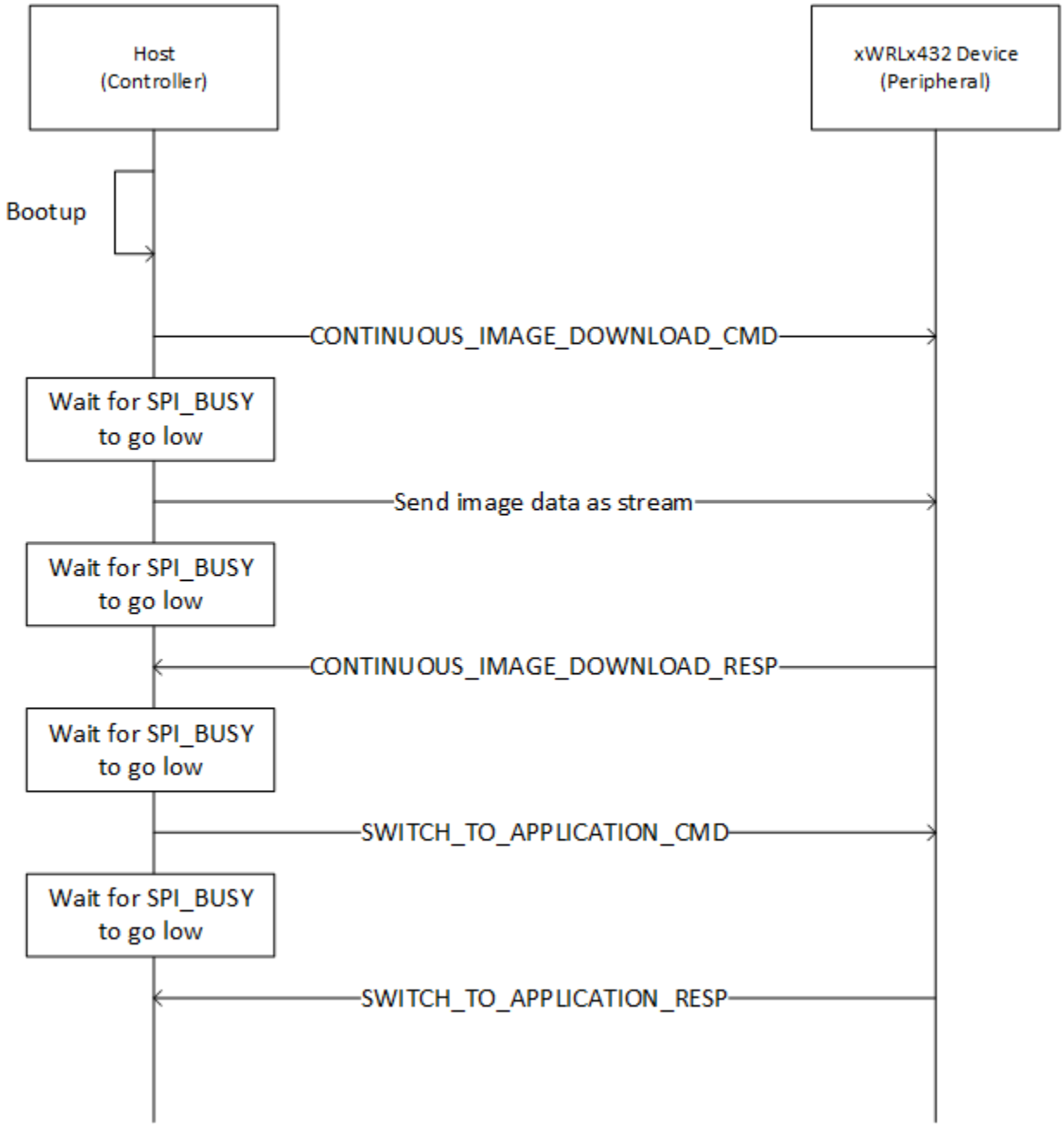


Figure 4-8. SPI download sequence

The last chunk of file download will block SPI_BUSY(=HIGH) until the parser is not completely done. Because of this the last ack from device will contain the final parser status. For switching to application host needs to issue **SWITCH_TO_APPLICATION_CMD** command. Switch to application will be done once the **SWITCH_TO_APPLICATION_RESP** is completely sent out to host. Host should not attempt to send any RBL command after this. Before switching to application, RBL will block SPI_BUSY(=HIGH) which can eventually be unblocked (SPI_BUSY = LOW) by the application after bootup for communication with the host.

Note – after sending final ack, protocol layer will unblock SPI_BUSY (= LOW) as part of the general CMD-RESP sequence. Hence there is a small window before RBL can block the SPI_BUSY before switching to application, which needs to be ignored by the host device.

4.6.2.3 SPI Timing and Restrictions

The ratio of SPI serial clock speed and RBL M4 speed shall be maintained minimum 1:4, i.e the RBL M4 is running at 160MHz and SPI host serial clock shall not be greater than 40MHz.

SPI Serial Clock	Typical Image Download Time with DMA Implementation at HOST for 1kB Image
10MHz	~1ms
11MHz	~920us
20MHz	~520us
22MHz	~480us

4.6.2.4 Bootloader Commands

4.6.2.4.1 List of SPI Boot APIs

API Name	CMD_TYPE	Description
GET_RBL_STATUS_CMD	0x0010	This is a optional API to read the RBL status when control is with RBL, this API can be issued after reception of **CONTINUOUS_IMAGE_DOWNLOAD_RESP.**
GET_RBL_STATUS_RESP	0x0011	Response to **GET_RBL_STATUS_CMD.** The host shall wait for SPI_BUSY flag to go low to read the response from RBL.
CONTINUOUS_IMAGE_DOWNLOAD_CMD	0x0018	The Continuous image download API, is a 2 stage command-response protocol to speedup the image download using DMA from HOST. In stage 1, **CONTINUOUS_IMAGE_DOWNLOAD_CMD** command is sent to notify the RBL to setup DMA for continuous image download. In stage 2, after SPI_BUSY goes low, the host can pump the entire image using DMA as per image size configured in command.
CONTINUOUS_IMAGE_DOWNLOAD_RESP	0x0019	Response to **CONTINUOUS_IMAGE_DOWNLOAD_CMD.** The host shall wait for SPI_BUSY flag to go low to read the response from RBL.
SWITCH_TO_APPLICATION_CMD	0x001A	Command to switch to application, once this command is issued, the RBL will not respond to any of the SPI commands as per these protocols.
SWITCH_TO_APPLICATION_RESP	0x001B	Response to **SWITCH_TO_APPLICATION_CMD.** The switch to application will occur only after reading this response by Host.

4.6.2.5 Reference

4.6.2.5.1 GET_RBL_STATUS_CMD

Field Name	Number of bytes	Description
MSG_CRC	4	Ethernet CRC-32 standard. Polynomial: x32+x26+x23+x22+x16+x12+ x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

ADVANCE INFORMATION

Field Name	Number of bytes	Description
SPI_CMD_TYPE	2	0x0010
LONG_MSG_SIZE	2	0x0000
RESERVED	4	0x00000000
SHORT_MSG	4	0x00000000
LONG_MSG	0	NA

4.6.2.5.2 GET_RBL_STATUS_RESP

Field Name	Number of bytes	Description
MSG_CRC	4	Ethernet CRC-32 standard.
SPI_CMD_TYPE	2	0x0011
LONG_MSG_SIZE	2	0x0010
RESERVED	4	0x0000
SHORT_MSG	4	Error Code. This field will be populated by RBL with error codes, if any error in command parsing.

Error Code. This field will be populated by RBL with error codes, if any error in command parsing.

Error Code	Description
0x0000	NO_ERROR Command is Successful.
0x0011	INVALID_LONG_MSG_SIZE

RBL Status:

Field Name	Number of Bytes	Description
BOOT_ERROR_STATUS	8	Refer **RBL_ERROR_STATUS** section
RESERVED	8	RESERVED

4.6.2.5.3 CONTINUOUS_IMAGE_DOWNLOAD_CMD

This is a 2 stage command, this command followed by the actual image data which starts after SPI_BUSY=HIGH to LOW transition after sending this command.

Note: During image download, peripheral will transmit 0xF0F0 on Tx (CIPO)

Note: In case of any errors in the command, the RBL will not bring SPI_BUSY to state low, host shall reset the device to recover from this error.

Field Name	Number of bytes	Description
MSG_CRC	4	Ethernet CRC-32 standard. Polynomial: x32+x26+x23+x22+x16+x12+ x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1
SPI_CMD_TYPE	2	0x0018
LONG_MSG_SIZE	2	0x0010
RESERVED	4	0x0000
SHORT_MSG	4	0x0000
LONG_MSG	16	Image Size information.

Field Name	Number of Bytes	Description
SPI_DOWNLOAD_SIZE	4	SPI_DOWNLOAD_SIZE_IN_BYTES = META_IMAGE_SIZE + padding for FIFO_LEVEL multiple. FIFO_LEVEL is 16 bytes in RBL
META_IMAGE_SIZE	4	META_IMAGE_SIZE in bytes
RESERVED	8	RESERVED

4.6.2.5.4 CONTINUOUS_IMAGE_DOWNLOAD_CMD

Field Name	Number of bytes	Description
MSG_CRC	4	Ethernet CRC-32 standard.
SPI_CMD_TYPE	2	0x0019
LONG_MSG_SIZE	2	0x0010
RESERVED	4	0x0000
SHORT_MSG	4	0x00000000
LONG_MSG	16	

Status Response of the RBL.

Field Name	Number of Bytes	Description
BOOT_ERROR_STATUS	8	Refer **RBL_ERROR_STATUS** section
RESERVED	8	RESERVED

4.6.2.5.5 SWITCH_TO_APPLICATION_CMD

Field Name	Number of bytes	Description
MSG_CRC	4	Ethernet CRC-32 standard. Polynomial: x32+x26+x23+x22+x16+x12+ x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1
SPI_CMD_TYPE	2	0x001A
LONG_MSG_SIZE	2	0x0000
RESERVED	4	0x0000
SHORT_MSG	4	0x0000
LONG_MSG	0	NA

4.6.2.5.6 SWITCH_TO_APPLICATION_CMD

Field Name	Number of bytes	Description
MSG_CRC	4	Ethernet CRC-32 standard.
SPI_CMD_TYPE	2	0x001B
LONG_MSG_SIZE	2	0x0000
RESERVED	4	0x0000
SHORT_MSG	4	
LONG_MSG	0	NA

Error Code. This field will be populated by RBL with error codes, if any error in command parsing.

Error Code	Description
0x0000	NO_ERROR Command is Successful.
0x0011	INVALID_LONG_MSG_SIZE

Error Code	Description
0x0015	SWITCH_TO_APPLICATION_CMD_FAILURE

4.6.2.5.7 SWITCH_TO_APPLICATION_CMD

Error Code Bit Definition (8 bytes)	Error Description	Possible Causes
Bit 0	RPRC Image1 Authentication Failure	
Bit 1	RPRC Image2 Authentication Failure	
Bit 2	RPRC Image3 Authentication Failure	
Bit 3	RPRC Image4 Authentication Failure	
Bit 4	RPRC Header not found	
Bit 5	Metaheader not found	
Bit 6	RPRC file length mismatch	
Bit 7	RPRC Application file offset mismatch	
Bit 8	RPRC FEC file offset mismatch	
Bit 9	RPRC Invalid fields	
Bit 10	Application image not found	
Bit 11	Metaheader number of files error	
Bit 12	Metaheader CRC failure	
Bit 13	RPRC config file offset mismatch	
Bit 14	QSPI read time out	
Bit 15	Invalid shared memory configuration	
Bits (40:16)	Reserved	
Bit 41	UART invalid target memory	• SRAM storage type command used in SOP device management • SFLASH storage type command used in SOP functional mode
Bit 42	UART download incomplete	Full chunks not received
Bit 43	UART SFLASH download CRC mismatch	Calculated CRC did not match with expect CRC in flashing
Bits (44:63)	Reserved	

4.6.3 Booting Over UART

4.6.3.1 UART Communication Protocol

When not using a QSPI flash device to store the application image, the bootloader will look for an image to be loaded via SPI or UART interface, if the first ping/command is initiated through UART interface then RBL switch to UART mode of image download.

The UART communication protocol involves a simple command-response flow. The host first sends the command packet to the xWRLx432 device and then waits for a response from the device. Multi-byte fields are transmitted in big endian format with MSB bytes transmitted followed by the LSB bytes. Each command packet is immediately processed by the bootloader without any scheduling or processing in the background and then a valid response packet is sent. The response can either be an ACK, a NACK or it may contain some specific response information. Each command has a known response packet and those are documented in sections Bootloader Commands and Bootloader Commands.

4.6.3.2 Bootloader Commands

Ping Command

The ping command can be issued by the host for verifying the UART connection with the device.

Get Status Command

The get status command can be issued by the host to see the error code associated with the previous command. This can only be issued after receiving a NACK response from a command. The error codes are listed below:

Error Code Bits Definition (8 bytes)	Error Description	Possible Causes
Bits (40:0)	Reserved	
Bit 41	UART invalid target memory	1. SRAM storage type command used in SOP device management 2. SFLASH storage type command used in SOP functional mode
Bit 42	UART download incomplete	Full chunks not received
Bit 43	UART SFLASH download CRC mismatch	Calculated CRC did not match with expect CRC in flashing
Bits (44:63)	Reserved	

Get Version Command

This command is used to get the version of the ROM bootloader. The ROM version is 4 bytes of information:

1. byte[0] = Build version
2. byte[1] = Minor version
3. byte[2] = Major version
4. byte[3] = ROM Gen version

Open Download Command

The open download command is issued to open a file download context for either the serial flash download or SRAM download. File size, storage type and file type for the download information are all needed to send the open download command. The last 4 bytes of this command are reserved. Multi-byte fields are transmitted in big endian format with MSB bytes transmitted followed by the LSB bytes.

FILE_SIZE	4 bytes	size of image
STORAGE_TYPE	4 bytes	0x2 = Serial flash 0x4 = SRAM
FILE_TYPE	4 bytes	Note - this field is read only if STORAGE_TYPE is serial flash (0x2) 0x4 = META_IMAGE1 (0KB offset) 0x5 = META_IMAGE2 (512KB offset) 0x6 = META_IMAGE3 (1024KB offset) 0x7 = META_IMAGE4 (1536KB offset)

Send SFLASH Download Chunk Command

This command is used to send the image chunks after the "open download" command. Multi-byte fields are transmitted in big endian format with MSB bytes transmitted followed by the LSB bytes.

Send SRAM Download Chunk Command

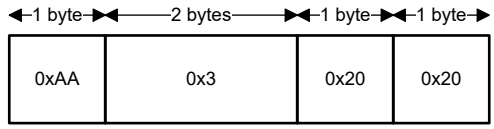
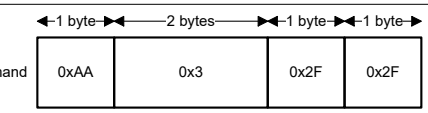
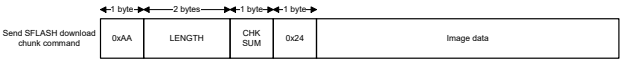
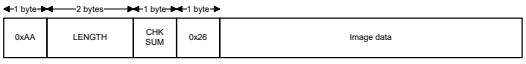
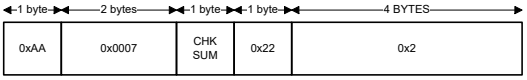
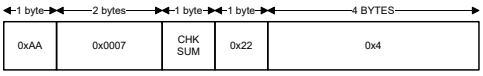
This command is used to send the image chunks after the "open download" command. Multi-byte fields are transmitted in big endian format with MSB bytes transmitted followed by the LSB bytes.

Close Download Command

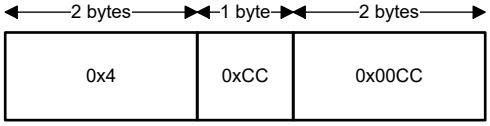
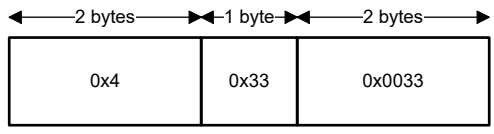
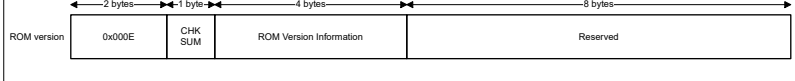
This command is used to close the download context for SFLASH. The payload is the storage type:

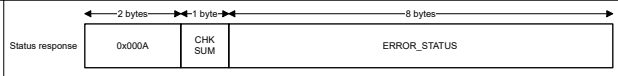
STORAGE_TYPE	4 bytes	0x2 = Serial flash 0x4 = SRAM

Command Structure

Packet	Length	Opcode	Structure	Expected Response	Expected Response Size
Ping	3	0x20		ACK/NACK	Payload of 4 bytes
Get Status	3	0x23		Status	Payload of 8 bytes
Get Version	3	0x2F		ROM Version	Payload of 12 bytes
Open Download	19	0x21		ACK/NACK	Payload of 4 bytes
Send SFLASH Download Chunk	Size of image data + 3	0x24		ACK/NACK	Payload of 4 bytes
Send SRAM Download Chunk	Size of image data + 3	0x26		ACK/NACK	Payload of 4 bytes
Close Download (SFLASH)	7	0x22		ACK/NACK	Payload of 4 bytes
Close Download (SRAM)	7	0x22		ACK/NACK	Payload of 4 bytes

Response Structure

Packet	Length	Structure	Response
ACK	4		
NACK	4		
ROM Version	14		byte[0] = Build versionbyte[1] = Minor versionbyte[2] = Major versionbyte[3] = ROM Gen version

Packet	Length	Structure	Response
Status	10		0 = no error See error codes in get status command description above

4.6.3.3 Example Host Application Flow

For an example command sequence, refer to Example Command Sequence. For functional mode, the open file command specifies SRAM as the target instead of SFLASH. Write commands for SRAM and SFLASH are different. Otherwise it's the same sequence for writing to SFLASH in flashing mode or writing to SRAM in functional mode.

4.6.4 Append Mode of Operation

In append mode of operation, ROM and RAM co-exist after the application boot up. In this mode, the reset types mentioned below will lead to program flow re-entering ROM upon reset. Because of this, RBL will need the AON registers to be preconfigured by application to pass control back to the application depending upon the reset cause.

4.6.4.1 Overview of AON Fields Used by Bootloader

The AON Registers fields which are used by bootloader in warm reset and deep sleep exit as shown in the image below.

The PC Register (PCR) 1 to 5 are used by bootloader ROM and the PCR 8 is used by FECSS, the PCR 6 and 7 can be used by Application.

Details

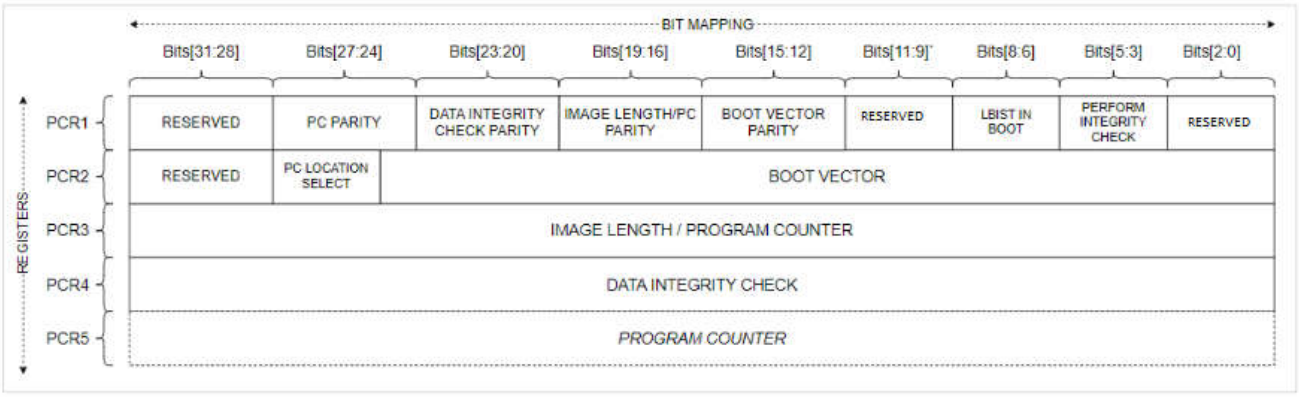


Figure 4-9. Bootloader Flow Details

Multibit Fields

The Multibit fields have a single bit information stored in 3 bits space for redundancy.

0x7 =>TRUE and 0x0 =>FALSE. If any other value is received then more 1s than 0s means true and more 0s than 1s means false

The following 5 multibit fields are used by the bootloader:

- **PERFORM INTEGRITY CHECK(PC REGISTER1<5:3>) :** This field will be used by bootloader in case the device has woken up from Warm Reset. It set to true, the RBL performs the integrity check on the Application Image present in the RAM memory.
  - The starting address for CRC calculation (integrity check) for application image is taken from the BOOT VECTOR field
  - The Image length for CRC Calculation is taken from IMAGE_LENGTH field
  - The expected CRC is read from DATA INTEGRITY CHECK field

ADVANCE INFORMATION

- **LBIST IN BOOT (PC REGISTER1<8:6>) :** This field will be used by the bootloader in case the device has woken up from Warm Reset. If set to true, LBIST test will be executed by the bootloader.
- **PC LOCATION SELECT(PC REGISTER2<27:25>) :** This field is used in case of Deep Sleep exit. The Program Counter register location for jumping back to the application can be stored in PC Register 3 or PC Register 5. If this field is set to true, the Program Counter will be stored at PC_REGISTER 5, otherwise it will be stored in PC_REGISTER 3

32 bit Fields

These fields contain 32 bit register information used by boot loader. Each of these fields have corresponding 4 bit parity field

- **BOOT VECTOR(PC REGISTER2<24:0>) :** This fields is the right shifted by 7 version of Boot Vector for application image. The boot vector will be stored by the bootloader during the cold boot and by the user in development mode. In warm reset scenario, bootloader will read the Boot Vector field to get the Application start address. 4 bit Parity for this field (for right shifted by 7 version) will be stored in **Boot Vector Parity (PC_REGISTER1<15:12>)**.
- **PROGRAM COUNRER(PC REGISTER3<31:0> or PC REGISTER5<31:0>) :** This the Program counter location for the application after waking up from Deep Sleep. The location of this field be decided by PC LCOATION SELECT field. RBL will jump to this location after the device has woken from Deep-sleep. The 4 bit parity will either be stored at **IMAGE LENGTH/PC PARITY(PC_REGISTER1<29:16>)** or **PC PARITY(PC_REGISTER1<27:24>)**.
- **IMAGE LENGTH(PC REGISTER3<31:0>) :** This is the Image length for the application (starting from Boot Vector) for which the Data Integrity is calculated and stored in the DATA INTEGRITY CHECK field. This field will be used by the bootloader in case the device has come out of Warm Reset and the application integrity check has to be performed. The 4 bit parity is stored for this field at **IMAGE LENGTH/PC PARITY(PC_REGISTER1<29:16>)**
- **DATA INTEGRITY CHECK(PC REGISTER4<31:0>) :** The expected CRC value for the Application image (in case the data integrity check is enabled from PERFORM INTEGRITY CHECK field) in case the device has come out of Warm Reset is stored in this field. This field has to be populated by the Application before entering into Warm Reset. The 4 bit parity for this field also has to be calculated and stored at **DATA INTEGRITY CHECK PARITY(PC _REGISTER1<23:20>)**.

Calculating the 4 bit parity

The 4 bit parity is always calculated for 32 bits. In case of the Boot Vector, the upper 7 bits are kept as zero to calculate the 4 bit parity. The following approach is used to calculate the 4 bit parity

```
UINT32 w_32bitField = <32bit Field Value>;

    UINT8 c_parityVal = 0U;
    UINT8 c_bitIndex = 0U;

    for (c_bitIndex = 0U; c_bitIndex < 32U; c_bitIndex = c_bitIndex + 4U)
    {
        c_parityVal = c_parityVal ^ (UINT8)((w_32bitField >>c_bitIndex) & 0xFU);
    }
```

Reset Cause Registers

Please use the bits[3:0] for latest reset cause identification:

All the reset registers will be cleared by the bootloader. **The reset register values will be stored in BOOT_INFO_REG0[23:0]**	
APP_CTRL:APPSS_BOOT_INFO_REG0[3:0]	**Reset Reason Identification by bootloader** M_BOOT_RESET_REASON_PORZ (0x1U) M_BOOT_RESET_REASON_WARM (0x2U) M_BOOT_RESET_REASON_DEEPSLEEP (0x3U) M_BOOT_RESET_REASON_SOFT (0x4U) M_BOOT_RESET_REASON_STC_WARM (0x5U) M_BOOT_RESET_REASON_STC_PORZ (0x6U)
APP_CTRL:APPSS_BOOT_INFO_REG0[7:4]	TOP_PRCM:SYS_RST_CAUSE[2:0]

All the reset registers will be cleared by the bootloader. The reset register values will be stored in BOOT_INFO_REG0[23:0]	
APP_CTRL:APPSS_BOOT_INFO_REG0[15:7]	TOP_PRCM:RADAR_WAKEUP_STATUS[7:0]
APP_CTRL:APPSS_BOOT_INFO_REG0[23:16]	APP_RCM:RST_CAUSE[7:0]

4.6.4.2 Handling of Different Reset Scenarios by Bootloader

4.6.4.2.1 Deep Sleep Exit by Application

AON register fields used for this reset are listed below.

Register	Field	Description
PCR2	PC LOCATION SELECT	Whether PC value is stored by application in PCR3 or PCR5
PCR3	PROGRAM COUNTER	If PC LOCATION SET == 0b000 In this case PROGRAM COUNTER parity is stored in **IMAGE LENGTH/PC PARITY(PC_REGISTER1<29:16>)**
PCR5	PROGRAM COUNTER	If PC LOCATION SET == 0b111 In this case PROGRAM COUNTER parity is stored in **PC PARITY(PC_REGISTER1<27:24>)**

4.6.4.2.2 Warm Reset by Application

AON register fields used by this reset are listed below.

Register	Field	Description
PC REGISTER1	**PERFORM INTEGRITY CHECK**	This field will be used by boot loader in case the device has woken up from Warm Reset. It set to true, the RBL performs the integrity check on the Application Image present in the RAM memory. • The starting address for CRC calculation (integrity check) for application image is taken from the BOOT VECTOR field • The Image length for CRC Calculation is taken from IMAGE_LENGTH field • The expected CRC is read from DATA INTEGRITY CHECK field
PC REGISTER1	**LBIST IN BOOT**	This field will be used by the bootloader in case the device has woken up from Warm Reset. If set to true, LBIST test will be executed by the RBL(in case the wakeup is from warm reset)
PC REGISTER2	**BOOT VECTOR**	This fields is the Righted shifted by 7 version of Boot Vector for application image. The boot vector will be stored by the RBL during the cold boot and by the user in Development mode. In Warm reset scenario, RBL will read the Boot Vector field to get the Application start address. 4 bit Parity for this field (for Right shifted by 7 version) will be stored in **Boot Vector Parity (PC_REGISTER1<15:12>)**.
PC REGISTER3	**IMAGE LENGTH**	This is the Image length for the application (starting from Boot Vector) for which the Data Integrity is calculated and stored in the DATA INTEGRITY CHECK field. This field will be used by the RBL in case the device has come out of Warm Reset and the application integrity check has to be performed. The 4 bit parity is stored for this field at **IMAGE LENGTH/PC PARITY(PC_REGISTER1<29:16>)**

ADVANCE INFORMATION

Register	Field	Description
PC REGISTER4	**DATA INTEGRITY CHECK**	The expected CRC value for the Application image (in case the data integrity check is enabled from PERFORM INTEGRITY CHECK field) in case the device has come of Warm Reset is stored in this field. This field has to be populated by the Application before entering into Warm Reset. The 4 bit parity for this field also has to be calculated and stored at **DATA INTEGRITY CHECK PARITY(PC _REGISTER1<23:20>)**.
PC REGISTER5	**PROGRAM COUNTER**	Optional, if not used for program counter, then application can use this.
PC REGISTER6	**Application use case**	Can be used by Application use case.
PC REGISTER7	**Application use case**	Can be used by Application use case.
PC REGISTER8	**RESERVED**	Reserved by TI FECSS ROM.

4.6.4.2.3 Soft Reset by Application

BOOT VECTOR(PC REGISTER2<24:0>) : This fields is the Righted shifted by 7 version of Boot Vector for application image. The boot vector will be stored by the RBL during the cold boot and by the user in Development mode. In soft reset scenario, RBL will read the Boot Vector field to get the Application start address. 4 bit Parity for this field (for Right shifted by 7 version) will be stored in **Boot Vector Parity (PC_REGISTER1<15:12>)**.

4.7 Boot Image Format

The bootloader will expect and only accept an image properly formatted in the below RPRC format. Alignment bytes are added to the end of each section

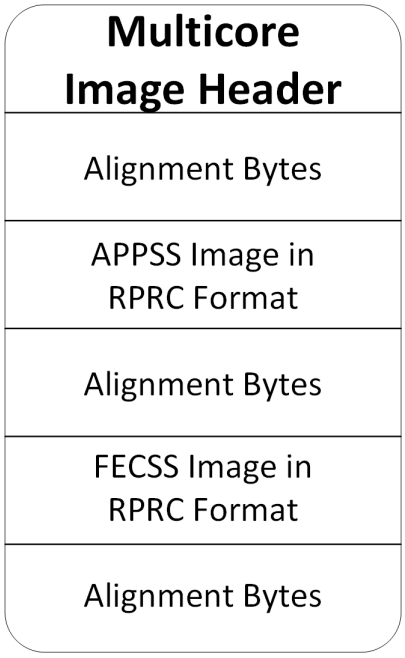


Figure 4-10. Multicore Image

The combined multicore Image header structure is listed below. There is support in the format for up to 5 images but only the first 2 are used for the xWRL6432.

Field	Size in bytes	Description
MSTR (magic number)	4	0x5254534D

Number of Images	4	Number of images present in the meta image. Max images supported 5.
Image Boot Mode(APPEND only)	4	Default value 0, other values not supported.
Header CRC	4	Meta Header CRC
Boot Vector	4	Start address of boot vector in RAM in Append mode.
Meta Image Size	4	Size of Meta Image.
Img1 File Type	4	NA, reserved for future use.
Img1 Magic word	4	Image1 magic word. Magic word for each images. APPSS Image: 0x3551 FEC Image: 0xB551
Img1 File offset	4	NA, reserved for future use.
Img1 CRC	4	Image1 CRC.
Reserved	4	Reserved
Img1 File Size	4	Size of the Image1
Reserved	8	Reserved
Img2 File Type	4	NA, reserved for future use.
Img2 Magic word	4	Image2 magic word.
Img2 File offset	4	NA, reserved for future use.
Img2 CRC	4	Image2 CRC.
Reserved	4	Reserved
Img2 File Size	4	Size of the Image2
Reserved	8	Reserved
..(Img3)	32	Image3 info (same as above)
..(Img4)	32	Image4 info (same as above)
Img5 File Type	4	NA, reserved for future use.
Img5 Magic word	4	Image5 magic word.
Img5 File offset	4	NA, reserved for future use.
Img5 CRC	4	Image5 CRC.
Reserved	4	Reserved
Img5 File Size	4	Size of the Image5
Reserved	8	Reserved
Shared Memory Allocation	4	Option to Enable APPSS Shared RAM allocation. 0b0 - Enable Shared ram bank 0 0b1 - Enable Shared ram bank 1
MEND	4	0x444E454D

4.8 Memory Mapped Debug Data for Application

4.8.1 Boot Information Store

```
/*!*****************************************************************************
 * @file export_bootinfo.h
 *
 * @brief Export File for Application
 *
 * @b Description @n
 * This header file contains the info for Boot Info strucutre. This file
 * can be used by the application to know the Boot status
 *
```

```
 * @note
 * <B> © Copyright 2020, Texas Instruments Incorporated - www.ti.com </B>
 * @n
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met:
 * @n
 * <ul>
 *   <li> Redistributions of source code must retain the above copyright
 *        notice, this list of conditions and the following disclaimer.
 * @n
 *   <li> Redistributions in binary form must reproduce the above copyright
 *        notice, this list of conditions and the following disclaimer in the
 *        documentation and/or other materials provided with the distribution.
 * @n
 *   <li> Neither the name of Texas Instruments Incorporated nor the names of
 *        its contributors may be used to endorse or promote products derived
 *        from this software without specific prior written permission.
 * </ul>
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS
 * IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
 * PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES LOSS OF USE,
 * DATA, OR PROFITS OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 ******************************************************************************
 */

/******************************************************************************
 * FILE INCLUSION PROTECTION
 ******************************************************************************
 */
#ifndef EXPORT_BOOTINFO_H
#define EXPORT_BOOTINFO_H

#ifdef __cplusplus
extern "C" {
#endif
/******************************************************************************
 * INCLUDE FILES
 ******************************************************************************
 */
#include <stdint.h>

/******************************************************************************
 * MACRO DEFINITIONS
 ******************************************************************************
 */

/*!
 * @brief Macro for different XTAL Frequencies
 */
#define M_XTAL_FREQ_25MHZ       (25000000U)
#define M_XTAL_FREQ_26MHZ       (26000000U)
#define M_XTAL_FREQ_38P4MHZ     (38400000U)
#define M_XTAL_FREQ_40MHZ       (40000000U)


/*!
 * @brief Memory Instance for MemInit Config
 */
#define M_MEM_INST_RAM_1        (0x01U)
#define M_MEM_INST_RAM_2        (0x02U)
#define M_MEM_INST_RAM_3        (0x04U)
#define M_MEM_INST_SHRD_RAM_0   (0x8U)
#define M_MEM_INST_SHRD_RAM_1   (0x10U)
#define M_MEM_INST_TPCC_A       (0x20U)
#define M_MEM_INST_FEC_RAM      (0x40U)
#define M_MEM_INST_FEC_SH_MEM   (0x80U)

/*!
 * @brief Memory groups for PBIST Low Register
```

```
 */
//changed
#define M_PBIST_MEMGRP_PBIST_ROM                (0x00000020U)
#define M_PBIST_MEMGRP_APPSS_ROM_BANK1          (0x00000001U)
#define M_PBIST_MEMGRP_APPSS_ROM_BANK2          (0x00000002U)
#define M_PBIST_MEMGRP_APPSS_BANK1_CLUSTER1     (0x00000800U)
#define M_PBIST_MEMGRP_APPSS_BANK1_CLUSTER3     (0x00001000U)
#define M_PBIST_MEMGRP_APPSS_BANK1_CLUSTER4     (0x00002000U)
#define M_PBIST_MEMGRP_APPSS_BANK2_CLUSTER2     (0x00004000U)
#define M_PBIST_MEMGRP_APPSS_BANK2_CLUSTER5     (0x00008000U)
#define M_PBIST_MEMGRP_APPSS_BANK3_CLUSTER5     (0x00010000U)
#define M_PBIST_MEMGRP_APPSS_SHMEM1_CLUSTER6    (0x00020000U)
#define M_PBIST_MEMGRP_APPSS_SHMEM2_CLUSTER6    (0x00040000U)
#define M_PBIST_MEMGRP_FECSS_BANK1_CLUSTER1     (0x80000000U)

/*!
 * @brief Memory groups for PBIST High Register
 */
#define M_PBIST_MEMGRP_FECSS_BANK2_CLUSTER2     (0x00000001U)
#define M_PBIST_MEMGRP_FECSS_SHMEM_CLUSTER3     (0x00000004U)

/*!
 * @brief Reset reasons identified by Boot
 */
#define M_BOOT_RESET_REASON_PORZ              (0x1U)
#define M_BOOT_RESET_REASON_WARM              (0x2U)
#define M_BOOT_RESET_REASON_DEEPSLEEP         (0x3U)
#define M_BOOT_RESET_REASON_SOFT              (0x4U)
#define M_BOOT_RESET_REASON_STC_WARM          (0x5U)
#define M_BOOT_RESET_REASON_STC_PORZ          (0x6U)

/*!
 * @brief Macro for different SOP Modes
 */
#define M_SOP_MODE_DEV_MGMT     (0x0U)
#define M_SOP_MODE_APPLICATION  (0x1U)
#define M_SOP_MODE_TEST         (0x2U)
#define M_SOP_MODE_DEBUG        (0x3U)

/*!
 * @brief Macro for PBIST status
 */
#define M_PBIST_STATUS_NOT_RUN                  (0x00U)
#define M_PBIST_STATUS_PASS                     (0x01U)
#define M_PBIST_STATUS_FAIL                     (0x02U)
#define M_PBIST_STATUS_TIME_OUT                 (0x03U)
/*!
 * @brief Macro for LBIST status
 */
#define M_LBIST_STATUS_NOT_RUN                  (0x00U)
#define M_LBIST_STATUS_PASS                     (0x01U)
#define M_LBIST_STATUS_FAIL                     (0x02U)


/*!
 * ESM Mapping
 *
 */
#define M_ESM_APPSS_RAM_ECC_AGG_UERR    (0x00000008U)
#define M_ESM_APPSS_RAM_ECC_AGG_SERR    (0x00000010U)
#define M_ESM_APPSS_TPCC_AGGR_ERR       (0x00100000U)

#define M_ESM_FECSS_RAM_ECC_AGG_UERR    (0x00000080U)
#define M_ESM_FECSS_RAM_ECC_AGG_SERR    (0x00000100U)
/*****************************************************************************
 * TYPE-DEFINE STRUCT/ENUM/UNION DEFINITIONS
 *****************************************************************************
 */
/**
 * @brief   typedef structure for BootInfo Object.
 *
 * @details
 * The structure is placeholder for Boot Info
 */
typedef struct
{
    /*!
     * Reserved Fields
```

```
 */
UINT32              Reserved1[2];

/*!
 * Bitmap Error Status
 */
/*
-------------------------------------------------------------------------------------------
Error Code Bits Definition (8 bytes)    Error Description
-------------------------------------------------------------------------------------------
Bit 0            -                      Parser Error RPRC Image1 Authentication Failure
Bit 1            -                      Parser Error RPRC Image2 Authentication Failure
Bit 2            -                      Parser Error RPRC  Image3 Authentication Failure
Bit 3            -                      Parser Error RPRC Image4 Authentication Failure
Bit 4            -                      Parser Error RPRC HDR_NOT_FOUND
Bit 5            -                      Parser Error Metaheader not found
Bit 6            -                      Parser Error RPRC file length mismatch
Bit 7            -                      Parser Error RPRC Application file offset mismatch
Bit 8            -                      Parser Error RPRC FEC file offset mismatch
Bit 9            -                      Parser Error RPRC Inavlid fields
Bit 10            -                      Parser Error Application image not found
Bit 11            -                      Pareser Error Metaheader number of files error
Bit 12            -                      Metaheader CRC failure
Bit 13            -                      RPRC config file offset mismatch
Bit 14            -                      QSPI read time out
Bit 15            -                      Invalid shared memory configuration
Bits (40:16)    -                      Reserved
Bit 41          -                      UART invalid target memory
Bit 42            -                      UART download incomplete
Bit 43            -                      UART SFLASH download CRC mismatch
Bits (44:63)    -                      Reserved
*/
UINT64              dw_ErrorStatus;

/*!
 * Reserved Fields
 */
UINT32              Reserved2[1];

/*!
 * Application Boot Vector Address
 * M_XTAL_FREQ_25MHZ
 * M_XTAL_FREQ_26MHZ
 * M_XTAL_FREQ_38P4MHZ
 * M_XTAL_FREQ_40MHZ
 */
UINT32              w_XtalFrequency;

/*!
 * Core Frequency
 */
UINT32              w_CoreFrequency;

    /*!
 * Bitmap for MemInit Config ( OR of multiple fields)
 * M_MEM_INST_RAM_1
 * M_MEM_INST_RAM_2
 * M_MEM_INST_RAM_3
 * M_MEM_INST_SHRD_RAM_0
 * M_MEM_INST_SHRD_RAM_1
 * M_MEM_INST_TPCC_A
 * M_MEM_INST_FEC_RAM
 * M_MEM_INST_FEC_SH_MEM
 */
UINT32              w_MeminitConfig;

/*!
 * Bitmap for PBIST Low Config ( OR of multiple fields)
 * M_PBIST_MEMGRP_PBIST_ROM
 * M_PBIST_MEMGRP_APPSS_ROM_BANK1
 * M_PBIST_MEMGRP_APPSS_ROM_BANK2
 * M_PBIST_MEMGRP_APPSS_BANK1_CLUSTER1
 * M_PBIST_MEMGRP_APPSS_BANK1_CLUSTER1
 * M_PBIST_MEMGRP_APPSS_BANK1_CLUSTER3
 * M_PBIST_MEMGRP_APPSS_BANK1_CLUSTER4
 * M_PBIST_MEMGRP_APPSS_BANK2_CLUSTER2
 * M_PBIST_MEMGRP_APPSS_BANK2_CLUSTER5
 * M_PBIST_MEMGRP_APPSS_BANK3_CLUSTER5
```

```
     * M_PBIST_MEMGRP_APPSS_SHMEM1_CLUSTER6
     * M_PBIST_MEMGRP_APPSS_SHMEM2_CLUSTER6
     * M_PBIST_MEMGRP_FECSS_BANK1_CLUSTER1
     */
    UINT32              w_PbistLowConfig;

    /*!
     * Bitmap for PBIST High Config ( OR of multiple fields)
     * M_PBIST_MEMGRP_FECSS_BANK2_CLUSTER2
     * M_PBIST_MEMGRP_FECSS_SHMEM_CLUSTER3
     */
    UINT32              w_PbistHighConfig;

    /*!
     * Reserved Fields
     */
    UINT32              w_Reserved3[5U];

    /*!
     * ESM aggregator Low Error Status for runtime errors
     * M_ESM_APPSS_RAM_ECC_AGG_SERR
     * M_ESM_FECSS_RAM_ECC_AGG_SERR
     * M_ESM_APPSS_TPCC_AGGR_ERR
     */
    UINT32              w_EsmErrorLowStatus;

    /*!
     * ESM aggregator High Error Status for runtime errors
     * M_ESM_APPSS_RAM_ECC_AGG_UERR
     * M_ESM_FECSS_RAM_ECC_AGG_UERR
     */
    UINT32              w_EsmErrorHighStatus;

    /*!
     * Boot Safety Configuration
     * 0 - Boot safety Disabled
     * 1 - Boot safety Enabled
     */
    UINT8               c_BootSafetyEnabled;

    /*!
     * ECC Disable for memory
     * 0 - ECC Enabled for RAMs
     * 1 - ECC Disabled for RAMs
     */
    UINT8               c_EccDisabled;

    /*!
     * Dig PLL Lock Status
     * 0 - DIG PLL Not Locked
     * 1 - DIG PLL Locked
     */
    UINT8               c_PllLocked;

    /*!
     * Reset reason
     * M_BOOT_RESET_REASON_PORZ
     * M_BOOT_RESET_REASON_WARM
     * M_BOOT_RESET_REASON_DEEPSLEEP
     * M_BOOT_RESET_REASON_SOFT
     * M_BOOT_RESET_REASON_STC_WARM
     * M_BOOT_RESET_REASON_STC_PORZ
     */
    UINT8               c_ResetReason;

    /*
     * Reserved
     */
    UINT8                 C_Reserved4[2];

    /*!
     * FEC Shared RAM config
     * 0 - Shared Memory for FEC is not available
     * 1 - Shared Memory for FEC is available
     */
    UINT8               c_IsFecShRamPresent;

    /*!
```

```
     * APP Shared RAM config
     * 0 - No shared mem  allocation for App
     * 1 - first 128KB bank is allocated to App
     * 2 - Second 128KB bank is allocated to App
     * 3 - Both 128KB banks are allocated to APP
     */
    UINT8               c_appShMemConfig;

    /*!
     * SOP Mode Config
     * M_SOP_MODE_DEV_MGMT
     * M_SOP_MODE_APPLICATION
     * M_SOP_MODE_TEST
     * M_SOP_MODE_DEBUG
     */
    UINT8               c_SopMode;

    /*!
     * Flash Status
     * 0 - Flash Exists
     * 1 - Flash Doesn't Exist
     */
    UINT8               c_FlashExists;

    /*
     * Reserved
     */
    UINT8               C_Reserved5[1];

    /*!
     * LBIST Enable Configuration
     */
    UINT8               c_LbistEnable;

    /*!
     * Field specifying the status of PBIST test for PBIST ROM
     * M_PBIST_STATUS_NOT_RUN
     * M_PBIST_STATUS_PASS
     * M_PBIST_STATUS_FAIL
     */
    UINT8               c_PbistPbistRomStatus;

    /*!
     * Field specifying the status of PBIST test for BOOT ROM
     * M_PBIST_STATUS_NOT_RUN
     * M_PBIST_STATUS_PASS
     * M_PBIST_STATUS_FAIL
     */
    UINT8               c_PbistBootRomStatus;

    /*!
     * Field specifying the status of PBIST test for DATA RAM
     * M_PBIST_STATUS_NOT_RUN
     * M_PBIST_STATUS_PASS
     * M_PBIST_STATUS_FAIL
     */
    UINT8               c_PbistDataRamStatus;

    /*!
     * Field specifying the status of PBIST test for LOADING RAM
     * M_PBIST_STATUS_NOT_RUN
     * M_PBIST_STATUS_PASS
     * M_PBIST_STATUS_FAIL
     * M_PBIST_STATUS_TIME_OUT
     */
    UINT8               c_PbistLoadRamStatus;

    /*!
     * Field specifying the status of LBIST Self Test
     * M_LBIST_STATUS_NOT_RUN
     * M_LBIST_STATUS_PASS
     * M_LBIST_STATUS_FAIL
     */
    UINT8               c_LbistSelfTestStatus;

    /*!
     * Field specifying the status of LBIST Test
     * M_LBIST_STATUS_NOT_RUN
```

```
     * M_LBIST_STATUS_PASS
     * M_LBIST_STATUS_FAIL
     */
    UINT8               c_LbistAppTestStatus;

    /*
     * Reserved Fields
     */
    UINT8               Reserved[7];
} T_BOOT_OBJECT;

/*********************  ************************************************************
 * EXTERN GLOBAL VARIABLES/DATA-TYPES DECLARATIONS
 ******************************************************************************
 */

#define M_BOOT_INFO_CONFIG     ((T_BOOT_OBJECT *)0x00458000U)

#ifdef __cplusplus
}
#endif

#endif /* EXPORT_BOOTINFO_H */
```

This header file contains the information for Boot Info(T_BOOT_OBJECT)Structure. This file can be used by the application to know the Boot status

Application shouldn't download to this RAM region because the boot ROM uses it.

Memory	Start address	Size	Remarks
Data memory	0x0045FB00	0x00008000	32KB

5 Device Configuration

5.1 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 Clock, Power and Reset Management in xWRLx432 is distributed. The Main subsystem TOPRCM module controls all the Subsystem Resets, Power and Clocks. The SubSystem RCM modules control their respective subsystem IPs The available address space of PRCM is divided as in Figure 5-1.

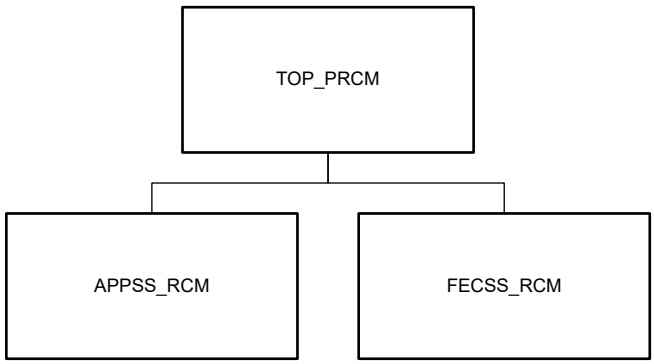


Figure 5-1. Device Configuration

Table 5-1. PRCM Space

PRCM Space	Description
TOP_PRCM	Top-level reset, power, clock management registers
APP_RCM	Application subsystem reset, clock management registers
FEC_RCM	Front-end Controller subsystem reset, clock management registers. This is handled by mmwave DFP APIs.

5.2 Control Registers

5.2.1 TOP_PRCM Registers

Table 5-2 lists the memory-mapped registers for the TOP_PRCM registers. All register offset addresses not listed in Table 5-2 should be considered as reserved locations and the register contents should not be modified.

Table 5-2. TOP_PRCM Registers

Offset	Acronym	Register Name	Section
0h	PID	PID register	Go
4h	CLK_REQ_PARAM		Go
8h	APP_PWR_REQ_PARAM		Go
Ch	FEC_PWR_REQ_PARAM		Go
10h	HWA_PWR_REQ_PARAM		Go
14h	APP_CORE_SYSRESET_PARAM		Go
18h	FEC_CORE_SYSRESET_PARAM		Go
1Ch	RELEASE_PAUSE		Go
20h	WU_COUNTER_END		Go
24h	WU_COUNTER_START		Go
28h	WU_COUNTER_PAUSE		Go
2Ch	GTS_COUNTER_END		Go
30h	SLEEP_COUNTER_END		Go
34h	WU_SOURCE_EN		Go
38h	UART_RTS_CLEAR		Go
3Ch	RADAR_WAKEUP_STATUS		Go
40h	RTC_COMPARE_LSB		Go
44h	RTC_COMPARE_MSB		Go
48h	RTC_COMPARE_EN		Go
4Ch	RTC_COUNT_LSB		Go
50h	RTC_COUNT_MSB		Go
54h	PC_REGISTER1		Go
58h	PC_REGISTER2		Go
5Ch	PC_REGISTER3		Go
60h	PC_REGISTER4		Go
64h	PC_REGISTER5		Go
68h	PC_REGISTER6		Go
6Ch	PC_REGISTER7		Go
70h	PC_REGISTER8		Go
74h	WAKEUP_IO_MUX_SEL		Go
78h	WAKEUP_INT_SOURCE_EN		Go
7Ch	PMS_POWER_MODE		Go
80h	PMS_SRAM_GO_TO_SLEEP_DELAY		Go
400h	PMS_SRAM_GO_TO_SLEEP_TIME		Go
404h	PMS_SRAM_WAKEUP_DELAY		Go
408h	PMS_SRAM_WAKEUP_TIME		Go
40Ch	PMS_SRAM_LDO_EN		Go
410h	PMS_SLEEP_NO_RTA_CFG		Go
414h	PMS_SRAM_LDO_TRIM		Go
418h	PMS_SRAM_KA_TRIM		Go
41Ch	PMS_DIG_GO_TO_SLEEP_DELAY		Go
420h	PMS_DIG_GO_TO_SLEEP_TIME		Go
424h	PMS_DIG_WAKEUP_DELAY		Go

Table 5-2. TOP_PRCM Registers (continued)

Offset	Acronym	Register Name	Section
428h	PMS_DIG_WAKEUP_TIME		Go
42Ch	PMS_DIG_LDO_EN		Go
430h	PMS_DIG_LDO_TRIM		Go
434h	PMS_DIG_KA_TRIM		Go
438h	PMS_PSCON_EN_WAIT_DELAY		Go
43Ch	PMS_BGAP_DIS_HIBERNATE		Go
440h	PMS_BGAP_DELAY1		Go
444h	PMS_BGAP_DELAY2		Go
448h	PMS_BGAP_EN_OVERRIDE		Go
44Ch	PMS_BGAP_CAP_OVERRIDE1		Go
450h	PMS_BGAP_CAP_OVERRIDE2		Go
454h	PSCON_APP_PD_EN		Go
458h	PSCON_FEC_PD_EN		Go
45Ch	PSCON_HWA_PD_EN		Go
460h	PSCON_TEST_DBG_PD_EN		Go
464h	PSCON_APP_PD_RAM_STATE		Go
468h	PSCON_APP_PD_RAM_GRP1_STATE		Go
46Ch	PSCON_APP_PD_RAM_GRP2_STATE		Go
470h	PSCON_HWA_PD_RAM_GRP3_STATE		Go
474h	PSCON_FEC_PD_RAM_STATE		Go
478h	PSCON_FEC_PD_RAM_GRP4_STATE		Go
47Ch	PSCON_RAM_FORCE_SWITCH_EN		Go
480h	PSCON_RAM_SWITCH_EN_OVERRIDE1		Go
484h	PSCON_RAM_SWITCH_EN_OVERRIDE2		Go
488h	PSCOM_RAM_SWITCH_DELAY		Go
48Ch	PSCON_DFTRTA_OVERRIDE		Go
490h	PSCON_VNWA_SWITCH_EN1		Go
494h	PSCON_VNWA_SWITCH_EN2		Go
498h	PSCON_SRAM_LDO_WEAK_PROCESS		Go
49Ch	CLKM_OSC_CLK_REQ		Go
4A0h	CLKM_OVERRIDE1		Go
4A4h	CLKM_OVERRIDE2		Go
4A8h	CLKM_OVERRIDE3		Go
4ACh	CLKM_OUTPUT_HOST_CLK_REQ_OVERRIDE		Go
4B0h	CLKM_SEL_OV_XT_DRIVE		Go
4B4h	CLKM_DELAY1		Go
4B8h	CLKM_DELAY2		Go
4BCh	CLKM_HOST_CLK_REQ_DELAY		Go
4C0h	RST_OVERRIDE		Go
4C4h	RST_SOFT_RESET		Go
4C8h	RST_WDT_RESET_EN		Go
4CCh	RST_APP_PD_SOFT_RESET		Go
4D0h	RST_FEC_PD_SOFT_RESET		Go
4D4h	RST_HWA_PD_SOFT_RESET		Go
4D8h	RST_SOFT_APP_CORE_SYSRESET_REQ		Go

Table 5-2. TOP_PRCM Registers (continued)

Offset	Acronym	Register Name	Section
4DCh	RST_SOFT_FEC_CORE_SYSRESET_REQ		Go
4E0h	SYS_RST_CAUSE		Go
4E4h	DUBUGSS_DISABLE		Go
4E8h	SLOW_CLK_CLKCTL		Go
4ECh	DEBUGSS_CLK_CLKCTL		Go
4F0h	EFUSE_10M_OSC_DISABLE		Go
4F4h	DEBUGSS_CLK_AUTOSWITCH		Go
4F8h	RADAR_SAFTY_ERROR_REG		Go
4FCh	RELEASEFROMWIR_REG		Go
500h	HWA_PD_MEM_SHARE_REG		Go
504h	FRC_OSC_CLK_GATE		Go
508h	MEMSWAP_REG		Go
50Ch	LIMP_MODE_STATUS		Go
510h	RTI_CLOCK_GATE_SLEEP_STATE		Go
514h	TOP_3318_LDO_EN_CTRL		Go
518h	RFANA_TOP_LDO_EN		Go
51Ch	RFANA_TOP_ISO_CTRL		Go
520h	CLK_CTRL_REG1_LDO_CLKTOP		Go
524h	CLK_CTRL_REG1_XO_SLICER		Go
528h	TOP_LDO_3318_CTRL_REG0		Go
52Ch	TOP_LDO_3318_CTRL_REG1		Go
530h	TOP_LDO_DIG_CTRL_REG0		Go
534h	TOP_LDO_DIG_CTRL_REG1		Go
538h	TOP_LDO_SRAM_CTRL_REG0		Go
53Ch	TOP_LDO_SRAM_CTRL_REG1		Go
1008h	LOCK0_KICK0	- KICK0 component	Go
100Ch	LOCK0_KICK1	- KICK1 component	Go
1010h	intr_raw_status	Interrupt Raw Status/Set Register	Go
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	Go
1018h	intr_enable	Interrupt Enable register	Go
101Ch	intr_enable_clear	Interrupt Enable Clear register	Go
1020h	eoi	EOI register	Go
1024h	fault_address	Fault Address register	Go
1028h	fault_type_status	Fault Type Status register	Go
102Ch	fault_attr_status	Fault Attribute Status register	Go
1030h	fault_clear	Fault Clear register	Go
1800h	DEBUG_LOGIC_PSCON_OVERRIDE_SEL		Go
1804h	DEBUG_LOGIC_PSCON_OVERRIDE_VAL		Go
1808h	DEBUG_MEM_PSCON_OVERRIDE_SEL_1		Go
180Ch	DEBUG_MEM_PSCON_OVERRIDE_SEL_2		Go
1810h	DEBUG_MEM_PSCON_OVERRIDE_VAL_1		Go
1814h	DEBUG_MEM_PSCON_OVERRIDE_VAL_2		Go

Table 5-2. TOP_PRCM Registers (continued)

Offset	Acronym	Register Name	Section
1C00h	MCUCLKOUT_CLKCTL		Go
1C04h	MCUCLKOUT_CLKSTAT		Go
1C08h	DCDC_CTRL_REG1		Go
1C0Ch	DCDC_CTRL_REG2		Go
1C10h	DCDC_CTRL_REG3		Go
1C14h	DCDC_SLOPE_REG		Go
1C18h	APP_CPU_CLKCTL		Go
1C1Ch	VNWA_SWITCH_SCREEN_ENABLE		Go
1C20h	PLLDIG_RST_SRC_SEL		Go
1C24h	MEM_POWERDOWN_ACCESS_ERR_DIS		Go
1C28h	MEM_SWAP		Go
1C2Ch	SPARE_REG		Go

Complex bit access types are encoded to fit into small table cells. Table 5-3 shows the codes that are used for access types in this section.

Table 5-3. TOP_PRCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

5.2.1.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in Table 5-4.

Return to the Summary Table.

PID register

Table 5-4. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

5.2.1.2 CLK_REQ_PARAM Register (Offset = 4h) [Reset = X]

CLK_REQ_PARAM is shown in Table 5-5.

Return to the Summary Table.

Table 5-5. CLK_REQ_PARAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	deepsleep_out_state	R/W	1h	clk_req in deep sleep state 0x 0 - Oscillator clk enable 0x 1 - Oscillator clk disable
16-13	go_to_sleep_delay	R/W	0h	Go to delay sleep delay
12	wakeup_out_state	R/W	1h	In manual mode, this bit will determine the clk_req state directly 0x 0 - Oscillator clk disable 0x 1 - Oscillator clk enable In auto mode, this bit determines clk_req state at the wakeup. it also depends on the Radar power state FSM and the wakeup delay
11	mode	R/W	1h	0x 0 - Manual mode 0x 1 - Auto mode
10-0	wakeup_delay_count	R/W	7FFh	Wakeup delay count

5.2.1.3 APP_PWR_REQ_PARAM Register (Offset = 8h) [Reset = X]

APP_PWR_REQ_PARAM is shown in Table 5-6.

Return to the Summary Table.

Table 5-6. APP_PWR_REQ_PARAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	deepsleep_out_state	R/W	1h	Domain power state in deep sleep state 0x 0 - Domain power up 0x 1 - Domain power down
16-13	go_to_sleep_delay	R/W	0h	Go to delay sleep delay
12	wakeup_out_state	R/W	1h	In manual mode, this bit will determine the power state of the domain directly 0x 0 - Domain power down 0x 1 - Domain power up. In auto mode, this bit determines power state of the domain at the wakeup. it also depends on the Radar power state FSM and the wakeup delay
11	mode	R/W	1h	0x 0 - Manual mode 0x 1 - Auto mode
10-0	wakeup_delay_count	R/W	7FFh	Wakeup delay count

5.2.1.4 FEC_PWR_REQ_PARAM Register (Offset = Ch) [Reset = X]

FEC_PWR_REQ_PARAM is shown in Table 5-7.

Return to the Summary Table.

Table 5-7. FEC_PWR_REQ_PARAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	deepsleep_out_state	R/W	1h	Domain power state in deep sleep state 0x 0 - Domain power up 0x 1 - Domain power down
16-13	go_to_sleep_delay	R/W	0h	Go to delay sleep delay

Table 5-7. FEC_PWR_REQ_PARAM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	wakeup_out_state	R/W	1h	In manual mode, this bit will determine the power state of the domain directly 0x 0 - Domain power down 0x 1 - Domain power up. In auto mode, this bit determines power state of the domain at the wakeup. it also depends on the Radar power state FSM and the wakeup delay
11	mode	R/W	1h	0x 0 - Manual mode 0x 1 - Auto mode
10-0	wakeup_delay_count	R/W	7FFh	Wakeup delay count

5.2.1.5 HWA_PWR_REQ_PARAM Register (Offset = 10h) [Reset = X]

HWA_PWR_REQ_PARAM is shown in Table 5-8.

Return to the Summary Table.

Table 5-8. HWA_PWR_REQ_PARAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	deepsleep_out_state	R/W	1h	Domain power state in deep sleep state 0x 0 - Domain power up 0x 1 - Domain power down
16-13	go_to_sleep_delay	R/W	0h	Go to delay sleep delay
12	wakeup_out_state	R/W	1h	In manual mode, this bit will determine the power state of the domain directly 0x 0 - Domain power down 0x 1 - Domain power up. In auto mode, this bit determines power state of the domain at the wakeup. it also depends on the Radar power state FSM and the wakeup delay
11	mode	R/W	1h	0x 0 - Manual mode 0x 1 - Auto mode
10-0	wakeup_delay_count	R/W	7FFh	Wakeup delay count

5.2.1.6 APP_CORE_SYSRESET_PARAM Register (Offset = 14h) [Reset = X]

APP_CORE_SYSRESET_PARAM is shown in Table 5-9.

Return to the Summary Table.

Table 5-9. APP_CORE_SYSRESET_PARAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	deepsleep_out_state	R/W	1h	In deep sleep state 0x 0 - Core reset released 0x 1 - Core in reset
16-13	go_to_sleep_delay	R/W	0h	Go to delay sleep delay
12	wakeup_out_state	R/W	1h	In manual mode, this bit will determine the core reset state 0x 0 - Core in reset 0x 1 - Core reset released In auto mode, this bit determines core reset state at the wakeup. it also depends on the Radar power state FSM and the wakeup delay

Table 5-9. APP_CORE_SYSRESET_PARAM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	mode	R/W	1h	0x 0 - Manual mode 0x 1 - Auto mode
10-0	wakeup_delay_count	R/W	7FFh	Wakeup delay count

5.2.1.7 FEC_CORE_SYSRESET_PARAM Register (Offset = 18h) [Reset = X]

FEC_CORE_SYSRESET_PARAM is shown in Table 5-10.

Return to the Summary Table.

Table 5-10. FEC_CORE_SYSRESET_PARAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	deepsleep_out_state	R/W	1h	In deep sleep state 0x 0 - Core reset released 0x 1 - Core in reset
16-13	go_to_sleep_delay	R/W	0h	Go to delay sleep delay
12	wakeup_out_state	R/W	0h	In manual mode, this bit will determine the core reset state 0x 0 - Core in reset 0x 1 - Core reset released In auto mode, this bit determines core reset state at the wakeup. it also depends on the Radar power state FSM and the wakeup delay
11	mode	R/W	0h	0x 0 - Manual mode 0x 1 - Auto mode
10-0	wakeup_delay_count	R/W	7FFh	Wakeup delay count

5.2.1.8 RELEASE_PAUSE Register (Offset = 1Ch) [Reset = X]

RELEASE_PAUSE is shown in Table 5-11.

Return to the Summary Table.

Table 5-11. RELEASE_PAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	release_pause	R/W	0h	This register can be used to disable the RADAR power state FSM after the wakeup of the device. 0x 0 -> After wakeup state counter will stop at wu_counter_pause value 0x 1 -> After wakeup state counter won't stop at wu_counter_pause value

5.2.1.9 WU_COUNTER_END Register (Offset = 20h) [Reset = X]

WU_COUNTER_END is shown in Table 5-12.

Return to the Summary Table.

Table 5-12. WU_COUNTER_END Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	X	

Table 5-12. WU_COUNTER_END Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-0	wu_counter_end	R/W	40h	After wakeup until the state counter reaches wu_counter_end RADAR power state FSM remains in the WAKEUP state. In this state the go to sleep commands have no effect on the RADAR power state.

5.2.1.10 WU_COUNTER_START Register (Offset = 24h) [Reset = X]

WU_COUNTER_START is shown in Table 5-13.

Return to the Summary Table.

Table 5-13. WU_COUNTER_START Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	X	
10-0	wu_counter_start	R/W	0h	After wakeup from sleep signal, the state counter starts from the wu_counter_start value. This register can be used if fast wakeup is required.

5.2.1.11 WU_COUNTER_PAUSE Register (Offset = 28h) [Reset = X]

WU_COUNTER_PAUSE is shown in Table 5-14.

Return to the Summary Table.

Table 5-14. WU_COUNTER_PAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	X	
10-0	wu_counter_pause	R/W	37h	The state counter will be paused at this value if the release_pause is deasserted

5.2.1.12 GTS_COUNTER_END Register (Offset = 2Ch) [Reset = X]

GTS_COUNTER_END is shown in Table 5-15.

Return to the Summary Table.

Table 5-15. GTS_COUNTER_END Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	gts_counter_end	R/W	3h	After go to deep sleep command, RADAR power state FSM will remain in the GO_TO_DEEP_SLEEP state until the state counter reaches this value. In this state wakeup source has no effect on the device power state.

5.2.1.13 SLEEP_COUNTER_END Register (Offset = 30h) [Reset = X]

SLEEP_COUNTER_END is shown in Table 5-16.

Return to the Summary Table.

Table 5-16. SLEEP_COUNTER_END Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-0	sleep_count_end	R/W	7D00h	Sleep counter is one of the wakeup source. After the sleep counter reaches this value wakeup signal is generated.

5.2.1.14 WU_SOURCE_EN Register (Offset = 34h) [Reset = X]

WU_SOURCE_EN is shown in Table 5-17.

Return to the Summary Table.

Table 5-17. WU_SOURCE_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	X	
11	syncin_io_edge	R/W	1h	1'b 0 : posedge selected for wakeup 1'b 1 : negedge selected for wakeup
10	gpio_int_edge	R/W	1h	1'b 0 : posedge selected for wakeup 1'b 1 : negedge selected for wakeup
9	spi_cs_edge	R/W	1h	1'b 0 : posedge selected for wakeup 1'b 1 : negedge selected for wakeup
8	uart_rx_edge	R/W	1h	1'b 0 : posedge selected for wakeup 1'b 1 : negedge selected for wakeup
7-6	RESERVED	R/W	X	
5-0	wu_source_en	R/W	3Fh	It selects the wakeup source from deep sleep/sleep state of the device Bit 0 -> Sleep counter Bit 1 -> UART RX Bit 2 -> SPI CS Bit 3 -> GPIO or SYNCIN IO depends on the wakeup_io_mux_sel register Bit 4 -> RTC counter Bit 5 -> FRC frame start intr (Use only for device SLEEP wakeup)

5.2.1.15 UART_RTS_CLEAR Register (Offset = 38h) [Reset = X]

UART_RTS_CLEAR is shown in Table 5-18.

Return to the Summary Table.

Table 5-18. UART_RTS_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	uart_rts_clear	R/W	0h	If the device wakeup source is uart then the uart rts pin is pulled low by register. This register is used to clear the pulled low rts register. Writing 1 to this register will clear the pulled low for the UART rts

5.2.1.16 RADAR_WAKEUP_STATUS Register (Offset = 3Ch) [Reset = X]

RADAR_WAKEUP_STATUS is shown in Table 5-19.

Return to the Summary Table.

Table 5-19. RADAR_WAKEUP_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20	radar_state_is_deep_sleep	R	0h	RADAR power FSM is in DEEP_SLEEP state
19	radar_state_is_go_to_deep_sleep	R	0h	RADAR power FSM is in GO_TO_DEEP_SLEEP state
18	radar_state_is_sleep	R	0h	RADAR power FSM is in SLEEP state

Table 5-19. RADAR_WAKEUP_STATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	radar_state_is_idle	R	0h	RADAR power FSM is in IDLE state
16	radar_state_is_wake_up	R	0h	RADAR power FSM is in WAKEUP state
15-9	RESERVED	R/W	X	
8	wakeup_status_clear	R/W	0h	Clear's the wakeup status and source register 0x 0 -> Wakeup status and source capture enable 0x 1 -> Wakeup status and source reg clear and disable
7-2	wakeup_source	R	0h	It indicate wakeup source from SLEEP/DEEP SLEEP state Bit 0 -> Sleep counter as Wakeup source Bit 1 -> UART as Wakeup source Bit 2 -> SPI as Wakeup source Bit 3 -> GPIO as Wakeup source Bit 4 -> RTC counter as Wakeup source Bit 5 -> FRC frame start intr as Wakeup source
1-0	wakeup_status	R	0h	It indicates the wakeup status of the device, whether it is Wakeup due to POR or from SLEEP/DEEP SLEEP state 0x 1 -> Wakeup from SLEEP 0x 2 -> Wakeup from DEEP SLEEP

5.2.1.17 RTC_COMPARE_LSB Register (Offset = 40h) [Reset = 00000000h]

RTC_COMPARE_LSB is shown in Table 5-20.

Return to the Summary Table.

Table 5-20. RTC_COMPARE_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	rtc_counter_compare_value_lsb	R/W	0h	48 bit RTC compare registers lsb 32 bit

5.2.1.18 RTC_COMPARE_MSB Register (Offset = 44h) [Reset = X]

RTC_COMPARE_MSB is shown in Table 5-21.

Return to the Summary Table.

Table 5-21. RTC_COMPARE_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	X	
15-0	rtc_counter_compare_value_msb	R/W	0h	48 bit RTC compare registers msb 16 bit

5.2.1.19 RTC_COMPARE_EN Register (Offset = 48h) [Reset = X]

RTC_COMPARE_EN is shown in Table 5-22.

Return to the Summary Table.

Table 5-22. RTC_COMPARE_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	rtc_counter_compare_enable_status	R	0h	RTC compare enable status from RTC module
0	rtc_counter_compare_enable	R/W	0h	RTC compare enable 0x 0 -> RTC compare disable 0x 1 -> RTC compare enable

5.2.1.20 RTC_COUNT_LSB Register (Offset = 4Ch) [Reset = 00000000h]

RTC_COUNT_LSB is shown in Table 5-23.

Return to the Summary Table.

Table 5-23. RTC_COUNT_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	rtc_counter_value_lsb	R	0h	48 bit RTC count registers lsb 32 bit

5.2.1.21 RTC_COUNT_MSB Register (Offset = 50h) [Reset = X]

RTC_COUNT_MSB is shown in Table 5-24.

Return to the Summary Table.

Table 5-24. RTC_COUNT_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	rtc_counter_value_msb	R	0h	48 bit RTC count registers msb 16 bit

5.2.1.22 PC_REGISTER1 Register (Offset = 54h) [Reset = 00000000h]

PC_REGISTER1 is shown in Table 5-25.

Return to the Summary Table.

Table 5-25. PC_REGISTER1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register1	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 1

5.2.1.23 PC_REGISTER2 Register (Offset = 58h) [Reset = 00000000h]

PC_REGISTER2 is shown in Table 5-26.

Return to the Summary Table.

Table 5-26. PC_REGISTER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register2	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 2

5.2.1.24 PC_REGISTER3 Register (Offset = 5Ch) [Reset = 00000000h]

PC_REGISTER3 is shown in Table 5-27.

Return to the Summary Table.

Table 5-27. PC_REGISTER3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register3	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 3

5.2.1.25 PC_REGISTER4 Register (Offset = 60h) [Reset = 00000000h]

PC_REGISTER4 is shown in Table 5-28.

Return to the Summary Table.

Table 5-28. PC_REGISTER4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register4	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 4

5.2.1.26 PC_REGISTER5 Register (Offset = 64h) [Reset = 00000000h]

PC_REGISTER5 is shown in Table 5-29.

Return to the Summary Table.

Table 5-29. PC_REGISTER5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register5	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 5

5.2.1.27 PC_REGISTER6 Register (Offset = 68h) [Reset = 00000000h]

PC_REGISTER6 is shown in Table 5-30.

Return to the Summary Table.

Table 5-30. PC_REGISTER6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register6	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 6

5.2.1.28 PC_REGISTER7 Register (Offset = 6Ch) [Reset = 00000000h]

PC_REGISTER7 is shown in Table 5-31.

Return to the Summary Table.

Table 5-31. PC_REGISTER7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register7	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 7

5.2.1.29 PC_REGISTER8 Register (Offset = 70h) [Reset = 00000000h]

PC_REGISTER8 is shown in Table 5-32.

Return to the Summary Table.

Table 5-32. PC_REGISTER8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	pc_register8	R/W	0h	Softwear registers to store 32 bit date in AOD domain register 8

5.2.1.30 WAKEUP_IO_MUX_SEL Register (Offset = 74h) [Reset = X]

WAKEUP_IO_MUX_SEL is shown in Table 5-33.

Return to the Summary Table.

Table 5-33. WAKEUP_IO_MUX_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	wakeup_io_mux_sel	R/W	0h	Mux select for the wakeup IO between GPIO and SYNCIN IO 0x 0 -> GPIO is selected 0x 1 -> SYNCIN_IO is selected

5.2.1.31 WAKEUP_INT_SOURCE_EN Register (Offset = 78h) [Reset = X]

WAKEUP_INT_SOURCE_EN is shown in Table 5-34.

Return to the Summary Table.

Table 5-34. WAKEUP_INT_SOURCE_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	radar_devicesleep_wakeup_interrupt_en	R/W	0h	Enable for the 'radar_devicesleep_wakeup_interrupt' during the device deep sleep exit, this will be always be generated during the device sleep exit 0x 0 -> 'radar_devicesleep_wakeup_interrupt' disable for deep sleep exit 0x 1 -> 'radar_devicesleep_wakeup_interrupt' enable for deep sleep exit
7-2	RESERVED	R/W	X	
1-0	wakeup_interrupt_source_en	R/W	3h	Source enable for the wakeup interrupt to CM4, Bit 0 : GPIO or SYNCIN_IO Bit 1 : RTC compare

5.2.1.32 PMS_POWER_MODE Register (Offset = 7Ch) [Reset = X]

PMS_POWER_MODE is shown in Table 5-35.

Return to the Summary Table.

Table 5-35. PMS_POWER_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	ldo_mode_status	R	0h	LDO mode 0x 0 -> 1.2V extenal supply is present, LDOs will be disable 0x 1 -> 1.2V External supply is not present, LDO will be used
15-4	RESERVED	R/W	X	
3	ov_pms_active_mode	R/W	0h	Override controls the Power state of the PMS FSMs i.e. SRAM LDO FSM, DIG LDO FSM and BGAP FSM. Can take these FSM forcefully to sleep state or active state. 0x 0 -> SRAM LDO, DIG LDO and BGAP FSM in sleep state 0x 1 -> SRAM LDO, DIG LDO and BGAP FSM in active state
2-0	sel_ov_pms_active_mode_safe	R/W	0h	Mux select controls the Power state of the PMS FSMs i.e. SRAM LDO FSM, DIG LDO FSM and BGAP FSM. Can take these FSM forcefully to sleep state or active state. 0x 0 -> selects the functional PMS power mode 0x 7 -> select the override value PMS power mode

5.2.1.33 PMS_SRAM_GO_TO_SLEEP_DELAY Register (Offset = 80h) [Reset = X]

PMS_SRAM_GO_TO_SLEEP_DELAY is shown in Table 5-36.

Return to the Summary Table.

Table 5-36. PMS_SRAM_GO_TO_SLEEP_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	sram_ldo_go_to_sleep_delay	R/W	2h	SRAM LDO FSM will stay in active state for this delay clock cycles before starting the sleep sequence.

5.2.1.34 PMS_SRAM_GO_TO_SLEEP_TIME Register (Offset = 400h) [Reset = X]

PMS_SRAM_GO_TO_SLEEP_TIME is shown in Table 5-37.

Return to the Summary Table.

Table 5-37. PMS_SRAM_GO_TO_SLEEP_TIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	sram_ldo_go_to_sleep_time	R/W	0h	This Delay extend the intermediate state in the sleep sequence. In this intermediate state both SRAM LDO and SRAM KA LDO are enable.

5.2.1.35 PMS_SRAM_WAKEUP_DELAY Register (Offset = 404h) [Reset = X]

PMS_SRAM_WAKEUP_DELAY is shown in Table 5-38.

Return to the Summary Table.

Table 5-38. PMS_SRAM_WAKEUP_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	sram_ldo_wakeup_delay	R/W	2h	SRAM LDO FSM will stay in sleep state for this delay clock cycles before starting the wakeup sequence.

5.2.1.36 PMS_SRAM_WAKEUP_TIME Register (Offset = 408h) [Reset = X]

PMS_SRAM_WAKEUP_TIME is shown in Table 5-39.

Return to the Summary Table.

Table 5-39. PMS_SRAM_WAKEUP_TIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	sram_ldo_wakeup_time	R/W	0h	This Delay extend the intermediate state in the wakeup sequence. In this intermediate state both SRAM LDO and SRAM KA LDO are enable.

5.2.1.37 PMS_SRAM_LDO_EN Register (Offset = 40Ch) [Reset = X]

PMS_SRAM_LDO_EN is shown in Table 5-40.

Return to the Summary Table.

Table 5-40. PMS_SRAM_LDO_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_sram_ka_ldo_en	R/W	0h	Override controls for the SRAM KA LDO enable irrespective of the SRAM LDO FSM state. 0x 0 -> SRAM KA LDO disable 0x 1 -> SRAM KA LDO enable
16	sel_ov_sram_ka_ldo_en	R/W	0h	Mux select controls for the SRAM KA LDO enable irrespective of the SRAM LDO FSM state. 0x 0 -> selects the SRAM LDO FSM output 0x 1 -> select the override value SRAM KA LDO enable
15-4	RESERVED	R/W	X	

Table 5-40. PMS_SRAM_LDO_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	ov_sram_ldo_en	R/W	0h	Override controls for the SRAM LDO enable irrespective of the SRAM LDO FSM state. 0x 0 -> SRAM LDO disable 0x 1 -> SRAM LDO enable
2-0	sel_ov_sram_ldo_en_safe	R/W	0h	Mux select controls for the SRAM LDO enable irrespective of the SRAM LDO FSM state. 0x 0 -> selects the SRAM LDO FSM output 0x 7 -> select the override value SRAM LDO enable

5.2.1.38 PMS_SLEEP_NO_RTA_CFG Register (Offset = 410h) [Reset = X]

PMS_SLEEP_NO_RTA_CFG is shown in Table 5-41.

Return to the Summary Table.

Table 5-41. PMS_SLEEP_NO_RTA_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	sram_ldo_common_rta_mode	R/W	0h	This signal shift the control of DFTRTA signals to the SRAM LDO FSM from PSCONs even in the non no-RTA-sleep state of SRA LDO FSM. 0x 0 -> select the DFTRTA signals from PSCONs 0x 1 -> select the DFTRTA signals from SRAM LDO FSM
0	sram_ldo_sleep_no_rta_mode	R/W	0h	It enables the no RTA sleep state for the SRAM LDO FSM. In no RTA sleep state the KA LDOs output can drop to 0x0.6V from 0x1.0x0V to reduce leakage power. 0x 0 -> SRAM LDO in RTA SLEEP state in DEEP SLEEP 0x 1 -> SRAM LDO in no RTA SLEEP state in DEEP SLEEP

5.2.1.39 PMS_SRAM_LDO_TRIM Register (Offset = 414h) [Reset = X]

PMS_SRAM_LDO_TRIM is shown in Table 5-42.

Return to the Summary Table.

Table 5-42. PMS_SRAM_LDO_TRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	X	
21-16	ov_sram_ldo_vtrim	R/W	0h	vtrim bits of SRAM LDO
15-1	RESERVED	R/W	X	
0	sel_ov_sram_ldo_vtrim	R/W	0h	It controls the SRAM LDO's vtrim bits source either from EFUSE or from registers 0x 0 -> selects vtrim bits from EFUSE 0x 1 -> selects vtrim bits from register overrides

5.2.1.40 PMS_SRAM_KA_TRIM Register (Offset = 418h) [Reset = X]

PMS_SRAM_KA_TRIM is shown in Table 5-43.

Return to the Summary Table.

Table 5-43. PMS_SRAM_KA_TRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	

Table 5-43. PMS_SRAM_KA_TRIM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29-24	ov_sram_ka_ldo_trim_no_rta	R/W	0h	vtrim bits of SRAM KA LDO in no RTA mode
23-22	RESERVED	R/W	X	
21-16	ov_sram_ka_ldo_trim_rta	R/W	0h	vtrim bits of SRAM KA LDO in RTA mode
15-1	RESERVED	R/W	X	
0	sel_ov_sram_ka_ldo_vtrim	R/W	0h	It controls the SRAM KA LDO's vtrim bits source either from EFUSE or from registers 0x 0 -> selects vtrim bits from EFUSE 0x 1 -> selects vtrim bits from register overrides

5.2.1.41 PMS_DIG_GO_TO_SLEEP_DELAY Register (Offset = 41Ch) [Reset = X]

PMS_DIG_GO_TO_SLEEP_DELAY is shown in Table 5-44.

Return to the Summary Table.

Table 5-44. PMS_DIG_GO_TO_SLEEP_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	dig_ldo_go_to_sleep_delay	R/W	2h	DIG LDO FSM will stay in active state for this delay clock cycles before starting the sleep sequence.

5.2.1.42 PMS_DIG_GO_TO_SLEEP_TIME Register (Offset = 420h) [Reset = X]

PMS_DIG_GO_TO_SLEEP_TIME is shown in Table 5-45.

Return to the Summary Table.

Table 5-45. PMS_DIG_GO_TO_SLEEP_TIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	dig_ldo_go_to_sleep_time	R/W	0h	This Delay extend the intermediate state in the sleep sequence. In this intermediate state both DIG LDO and DIG KA LDO are enable.

5.2.1.43 PMS_DIG_WAKEUP_DELAY Register (Offset = 424h) [Reset = X]

PMS_DIG_WAKEUP_DELAY is shown in Table 5-46.

Return to the Summary Table.

Table 5-46. PMS_DIG_WAKEUP_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	dig_ldo_wakeup_delay	R/W	2h	DIG LDO FSM will stay in sleep state for this delay clock cycles before starting the wakeup sequence.

5.2.1.44 PMS_DIG_WAKEUP_TIME Register (Offset = 428h) [Reset = X]

PMS_DIG_WAKEUP_TIME is shown in Table 5-47.

Return to the Summary Table.

Table 5-47. PMS_DIG_WAKEUP_TIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-0	dig_ldo_wakeup_time	R/W	0h	This Delay extend the intermediate state in the wakeup sequence. In this intermediate state both DIG LDO and DIG KA LDO are enable.

5.2.1.45 PMS_DIG_LDO_EN Register (Offset = 42Ch) [Reset = X]

PMS_DIG_LDO_EN is shown in Table 5-48.

Return to the Summary Table.

Table 5-48. PMS_DIG_LDO_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_dig_ka_ldo_en	R/W	0h	Override control for the DIG KA LDO enable state irrespective of the DIG LDO FSM state. 0x 0 -> DIG KA LDO disable 0x 1 -> DIG KA LDO enable
16	sel_ov_dig_ka_ldo_en	R/W	0h	Mux select control for the DIG KA LDO enable state irrespective of the DIG LDO FSM state. 0x 0 -> selects the DIG LDO FSM output 0x 1 -> select the override value DIG KA LDO enable
15-4	RESERVED	R/W	X	
3	ov_dig_ldo_en	R/W	0h	Override controls for the DIG LDO enable state irrespective of the DIG LDO FSM state. 0x 0 -> DIG LDO disable 0x 1 -> DIG LDO enable
2-0	sel_ov_dig_ldo_en_safe	R/W	0h	Mux select controls for the DIG LDO enable state irrespective of the DIG LDO FSM state. 0x 0 -> selects the DIG LDO FSM output 0x 7 -> select the override value DIG LDO enable

5.2.1.46 PMS_DIG_LDO_TRIM Register (Offset = 430h) [Reset = X]

PMS_DIG_LDO_TRIM is shown in Table 5-49.

Return to the Summary Table.

Table 5-49. PMS_DIG_LDO_TRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	X	
21-16	ov_dig_ldo_vtrim	R/W	0h	vtrim bits of DIG LDO
15-1	RESERVED	R/W	X	
0	sel_ov_dig_ldo_vtrim	R/W	0h	It controls the DIG LDO's vtrim bits source either from EFUSE or from registers 0x 0 -> selects vtrim bits from EFUSE 0x 1 -> selects vtrim bits from register overrides

5.2.1.47 PMS_DIG_KA_TRIM Register (Offset = 434h) [Reset = X]

PMS_DIG_KA_TRIM is shown in Table 5-50.

Return to the Summary Table.

Table 5-50. PMS_DIG_KA_TRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	X	
21-16	ov_dig_ka_ldo_vtrim	R/W	0h	vtrim bits of DIG KA LDO in RTA mode
15-1	RESERVED	R/W	X	
0	sel_ov_dig_ka_ldo_vtrim	R/W	0h	It controls the DIG KA LDO's vtrim bits source either from EFUSE or from registers 0x 0 -> selects vtrim bits from EFUSE 0x 1 -> selects vtrim bits from register overrides

5.2.1.48 PMS_PSCON_EN_WAIT_DELAY Register (Offset = 438h) [Reset = X]

PMS_PSCON_EN_WAIT_DELAY is shown in Table 5-51.

Return to the Summary Table.

Table 5-51. PMS_PSCON_EN_WAIT_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	X	
24	allow_pscon_en_before_dig_ldo_active	R/W	0h	It will make the assertion of the pscon_en dependent on the DIG LDO request but not on the DIG LFO FSM state
23-17	RESERVED	R/W	X	
16	pscon_en_force_active	R/W	0h	It bypasses the pscon_en_wait_delay and allows the power up of PSCONs irrespective of the DIG LDO FSM state or the DIG LDO request 0x 0 -> selects the functional pscon_en value 0x 1 -> select 0x1
15-8	RESERVED	R/W	X	
7-0	pscon_en_wait_delay	R/W	2h	It delays the power up of all power domains after the DIG LDO goes to active state. i.e. DIG LDO is enable and DIG KA LDO is disable

5.2.1.49 PMS_BGAP_DIS_HIBERNATE Register (Offset = 43Ch) [Reset = X]

PMS_BGAP_DIS_HIBERNATE is shown in Table 5-52.

Return to the Summary Table.

Table 5-52. PMS_BGAP_DIS_HIBERNATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	bgap_dis_hibernate	R/W	0h	It will keep the BGAP in active state even if the device is in deep sleep state. 0x 0 -> BGAP FSM can enter hibernate mode 0x 1 -> BGAP FSM's hibernate mode entry disable

5.2.1.50 PMS_BGAP_DELAY1 Register (Offset = 440h) [Reset = X]

PMS_BGAP_DELAY1 is shown in Table 5-53.

Return to the Summary Table.

Table 5-53. PMS_BGAP_DELAY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	X	
25-16	bgap_hib_refresh_time	R/W	280h	Controls the reference capacitor charging rate in the hibernate state

Table 5-53. PMS_BGAP_DELAY1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-7	RESERVED	R/W	X	
6-0	bgap_enter_sleep_delay	R/W	0h	BGAP FSM will stay in ACTIVE state for this delay clock cycles before starting the hibernate sequence.

5.2.1.51 PMS_BGAP_DELAY2 Register (Offset = 444h) [Reset = X]

PMS_BGAP_DELAY2 is shown in Table 5-54.

Return to the Summary Table.

Table 5-54. PMS_BGAP_DELAY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	X	
18-16	bgap_ref_cap_charge_time	R/W	0h	Controls the time duration for which the reference cap charging to be on
15-14	RESERVED	R/W	X	
13-8	bgap_pre_cap_charge_en_delay	R/W	Bh	Controls the time duration after which the external cap charging to be turned on
7-3	RESERVED	R/W	X	
2-0	bgap_cap_charge_time	R/W	6h	Controls the time duration for which the external cap charging to be on

5.2.1.52 PMS_BGAP_EN_OVERRIDE Register (Offset = 448h) [Reset = X]

PMS_BGAP_EN_OVERRIDE is shown in Table 5-55.

Return to the Summary Table.

Table 5-55. PMS_BGAP_EN_OVERRIDE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	ov_bgap_en	R/W	0h	Override control for the BGAP enable state irrespective of the BGAP FSM state. 0x 0 -> BGAP disable 0x 1 -> BGAP enable
2-0	sel_ov_bgap_en_safe	R/W	0h	Mux select control for the BGAP enable state irrespective of the BGAP FSM state. 0x 0 -> selects the BGAP FSM output 0x 7 -> select the override value BG_EN

5.2.1.53 PMS_BGAP_CAP_OVERRIDE1 Register (Offset = 44Ch) [Reset = X]

PMS_BGAP_CAP_OVERRIDE1 is shown in Table 5-56.

Return to the Summary Table.

Table 5-56. PMS_BGAP_CAP_OVERRIDE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_bgap_cap_charge_en	R/W	0h	Override control for the external cap charging irrespective of the BGAP FSM state. 0x 0 -> External cap charging disable 0x 1 -> External cap charging enable

Table 5-56. PMS_BGAP_CAP_OVERRIDE1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	sel_ov_bgap_cap_charge_en	R/W	0h	Mux select control for the external cap charging irrespective of the BGAP FSM state. 0x 0 -> selects the BGAP FSM output 0x 1 -> select the override value EN_CAP_CHARGE
15-2	RESERVED	R/W	X	
1	ov_bgap_cap_sw_enz	R/W	0h	Override control for the external cap switch state irrespective of the BGAP FSM state. 0x 0 -> External cap switch disable 0x 1 -> External cap switch enable
0	sel_ov_bgap_cap_sw_enz	R/W	0h	Mux select control for the external cap switch state irrespective of the BGAP FSM state. 0x 0 -> selects the BGAP FSM output 0x 1 -> select the override value BG_CAP_SW_ENZ

5.2.1.54 PMS_BGAP_CAP_OVERRIDE2 Register (Offset = 450h) [Reset = X]

PMS_BGAP_CAP_OVERRIDE2 is shown in Table 5-57.

Return to the Summary Table.

Table 5-57. PMS_BGAP_CAP_OVERRIDE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_bgap_hib_ref_cap_charge_en	R/W	0h	Override control for the reference cap charging irrespective of the BGAP FSM state. 0x 0 -> Reference cap charging disable 0x 1 -> Reference cap charging enable
16	sel_ov_bgap_hib_ref_cap_charge_en	R/W	0h	Mux select control for the reference cap charging irrespective of the BGAP FSM state. 0x 0 -> selects the BGAP FSM output 0x 1 -> select the override value
15-2	RESERVED	R/W	X	
1	ov_bgap_hib_cap_sw_en	R/W	0h	Override control for the reference cap switch state irrespective of the BGAP FSM state. 0x 0 -> Reference cap switch disable 0x 1 -> Reference cap switch enable
0	sel_ov_bgap_hib_cap_sw_en	R/W	0h	Mux select control for the reference cap switch state irrespective of the BGAP FSM state. 0x 0 -> selects the BGAP FSM output 0x 1 -> select the override value EN_HIB_CAP_SW_CTRL

5.2.1.55 PSCON_APP_PD_EN Register (Offset = 454h) [Reset = X]

PSCON_APP_PD_EN is shown in Table 5-58.

Return to the Summary Table.

Table 5-58. PSCON_APP_PD_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	

Table 5-58. PSCON_APP_PD_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	app_pd_reset_status	R	0h	Status bit for the APPSS power state 0x 0 -> APP PD POR reset released 0x 1 -> APP PD POR reset asserted
8	app_pd_power_status	R	0h	Status bit for the APPSS power state 0x 0 -> APP PD is power down 0x 1 -> APP PD is power up
7-2	RESERVED	R/W	X	
1	ov_app_pd_is_sleep	R/W	0h	Override control for the Power state of the APP PD. It can take APP PD to power down state or to power up state irrespective of the power request signals from the WAKE module. 0x 0 -> APP PD in active state 0x 1 -> APP PD in sleep state
0	sel_ov_app_pd_is_sleep	R/W	0h	Mux select control for the Power state of the APP PD. It can take APP PD to power down state or to power up state irrespective of the power request signals from the WAKE module. 0x 0 -> selects the functional sleep signal for APP PD 0x 1 -> select the override value sleep signal of APP PD

5.2.1.56 PSCON_FEC_PD_EN Register (Offset = 458h) [Reset = X]

PSCON_FEC_PD_EN is shown in Table 5-59.

Return to the Summary Table.

Table 5-59. PSCON_FEC_PD_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	
9	fec_pd_reset_status	R	0h	Status bit for the FECSS power state 0x 0 -> FEC PD POR reset released 0x 1 -> FEC PD POR reset asserted
8	fec_pd_power_status	R	0h	Status bit for the FECSS power state 0x 0 -> FEC PD is power down 0x 1 -> FEC PD is power up
7-2	RESERVED	R/W	X	
1	ov_fec_pd_is_sleep	R/W	0h	Override control for the Power state of the FEC PD. It can take FEC PD to power down state or to power up state irrespective of the power request signals from the WAKE module. 0x 0 -> FEC PD in active state 0x 1 -> FEC PD in sleep state
0	sel_ov_fec_pd_is_sleep	R/W	0h	Mux select control for the Power state of the FEC PD. It can take FEC PD to power down state or to power up state irrespective of the power request signals from the WAKE module. 0x 0 -> selects the functional sleep signal for FEC PD 0x 1 -> select the override value sleep signal of FEC PD

5.2.1.57 PSCON_HWA_PD_EN Register (Offset = 45Ch) [Reset = X]

PSCON_HWA_PD_EN is shown in Table 5-60.

Return to the Summary Table.

Table 5-60. PSCON_HWA_PD_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	

Table 5-60. PSCON_HWA_PD_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	hwa_pd_reset_status	R	0h	Status bit for the HWASS power state 0x 0 -> HWA PD POR reset released 0x 1 -> HWA PD POR reset asserted
8	hwa_pd_power_status	R	0h	Status bit for the HWASS power state 0x 0 -> HWA PD is power down 0x 1 -> HWA PD is power up
7-2	RESERVED	R/W	X	
1	ov_hwa_pd_is_sleep	R/W	0h	Override control for the Power state of the HWA PD. It can take HWA PD to power down state or to power up state irrespective of the power request signals from the WAKE module. 0x 0 -> HWA PD in active state 0x 1 -> HWA PD in sleep state
0	sel_ov_hwa_pd_is_sleep	R/W	0h	Mux select control for the Power state of the HWA PD. It can take HWA PD to power down state or to power up state irrespective of the power request signals from the WAKE module. 0x 0 -> selects the functional sleep signal for HWA PD 0x 1 -> select the override value sleep signal of HWA PD

5.2.1.58 PSCON_TEST_DBG_PD_EN Register (Offset = 460h) [Reset = X]

PSCON_TEST_DBG_PD_EN is shown in Table 5-61.

Return to the Summary Table.

Table 5-61. PSCON_TEST_DBG_PD_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	
9	test_dbg_pd_reset_status	R	0h	Status bit for the TEST DBG SS power state 0x 0 -> TEST DBG PD POR reset released 0x 1 -> TEST DBG PD POR reset asserted
8	test_dbg_pd_power_status	R	0h	Status bit for the TEST DBG PD power state 0x 0 -> TEST DBG PD is power down 0x 1 -> TEST DBG PD is power up
7-2	RESERVED	R/W	X	
1	ov_test_dbg_pd_is_sleep	R/W	0h	Override control for the Power state of the TEST DBG PD. It can be used to power down the TEST DBG PD to reduce the leakage and the active power. 0x 0 -> TEST DBG PD in active state 0x 1 -> TEST DBG PD in sleep state
0	sel_ov_test_dbg_pd_is_sleep	R/W	1h	Mux select control for the Power state of the TEST DBG PD. It can be used to power down the TEST DBG PD to reduce the leakage and the active power. 0x 0 -> selects the functional sleep signal for TEST DBG PD 0x 1 -> select the override value sleep signal of TEST DBG PD

5.2.1.59 PSCON_APP_PD_RAM_STATE Register (Offset = 464h) [Reset = X]

PSCON_APP_PD_RAM_STATE is shown in Table 5-62.

Return to the Summary Table.

Table 5-62. PSCON_APP_PD_RAM_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	X	

Table 5-62. PSCON_APP_PD_RAM_STATE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	app_pd_mem_active_state	R/W	7h	It controls which memory clusters in the APP PD needs to be powered up during the APP PD in power up state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in active state 0x 1 -> Memory cluster power on in active state
15-3	RESERVED	R/W	X	
2-0	app_pd_mem_sleep_state	R/W	0h	It controls which memory clusters in the APP PD needs to be powered up during the APP PD in power down state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in sleep state 0x 1 -> Memory cluster power on in sleep state

5.2.1.60 PSCON_APP_PD_RAM_GRP1_STATE Register (Offset = 468h) [Reset = X]

PSCON_APP_PD_RAM_GRP1_STATE is shown in Table 5-63.

Return to the Summary Table.

Table 5-63. PSCON_APP_PD_RAM_GRP1_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17-16	app_pd_mem_grp1_active_state	R/W	3h	It controls which memory clusters in the APP PD GROUP0x1 needs to be powered up during the APP PD in power up state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in active state 0x 1 -> Memory cluster power on in active state
15-2	RESERVED	R/W	X	
1-0	app_pd_mem_grp1_sleep_state	R/W	0h	It controls which memory clusters in the APP PD GROUP0x1 needs to be powered up during the APP PD in power down state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in sleep state 0x 1 -> Memory cluster power on in sleep state

5.2.1.61 PSCON_APP_PD_RAM_GRP2_STATE Register (Offset = 46Ch) [Reset = X]

PSCON_APP_PD_RAM_GRP2_STATE is shown in Table 5-64.

Return to the Summary Table.

Table 5-64. PSCON_APP_PD_RAM_GRP2_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	app_pd_mem_grp2_active_state	R/W	1h	It controls which memory clusters in the APP PD GROUP2 needs to be powered up during the APP PD in power up state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in active state 0x 1 -> Memory cluster power on in active state
15-1	RESERVED	R/W	X	
0	app_pd_mem_grp2_sleep_state	R/W	0h	It controls which memory clusters in the APP PD GROUP2 needs to be powered up during the APP PD in power down state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in sleep state 0x 1 -> Memory cluster power on in sleep state

5.2.1.62 PSCON_HWA_PD_RAM_GRP3_STATE Register (Offset = 470h) [Reset = X]

PSCON_HWA_PD_RAM_GRP3_STATE is shown in Table 5-65.

Return to the Summary Table.

Table 5-65. PSCON_HWA_PD_RAM_GRP3_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	X	
18-16	hwa_pd_mem_grp3_active_state	R/W	7h	It controls which memory clusters in the HWA PD needs to be powered up during the HWA PD in power up state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in active state 0x 1 -> Memory cluster power on in active state
15-3	RESERVED	R/W	X	
2-0	hwa_pd_mem_grp3_sleep_state	R/W	0h	It controls which memory clusters in the HWA PD needs to be powered up during the HWA PD in power down state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in sleep state 0x 1 -> Memory cluster power on in sleep state

5.2.1.63 PSCON_FEC_PD_RAM_STATE Register (Offset = 474h) [Reset = X]

PSCON_FEC_PD_RAM_STATE is shown in Table 5-66.

Return to the Summary Table.

Table 5-66. PSCON_FEC_PD_RAM_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	fec_pd_mem_active_state	R/W	1h	It controls which memory clusters in the FEC PD needs to be powered up during the FEC PD in power up state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in active state 0x 1 -> Memory cluster power on in active state
15-1	RESERVED	R/W	X	
0	fec_pd_mem_sleep_state	R/W	0h	It controls which memory clusters in the FEC PD needs to be powered up during the FEC PD in power down state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in sleep state 0x 1 -> Memory cluster power on in sleep state

5.2.1.64 PSCON_FEC_PD_RAM_GRP4_STATE Register (Offset = 478h) [Reset = X]

PSCON_FEC_PD_RAM_GRP4_STATE is shown in Table 5-67.

Return to the Summary Table.

Table 5-67. PSCON_FEC_PD_RAM_GRP4_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17-16	fec_pd_mem_grp4_active_state	R/W	3h	It controls which memory clusters in the FEC PD GROUP4 needs to be powered up during the FEC PD in power up state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in active state 0x 1 -> Memory cluster power on in active state
15-2	RESERVED	R/W	X	
1-0	fec_pd_mem_grp4_sleep_state	R/W	0h	It controls which memory clusters in the FEC PD GROUP4 needs to be powered up during the FEC PD in power down state. Each bit represent state of one cluster 0x 0 -> Memory cluster power off in sleep state 0x 1 -> Memory cluster power on in sleep state

5.2.1.65 PSCON_RAM_FORCE_SWITCH_EN Register (Offset = 47Ch) [Reset = X]

PSCON_RAM_FORCE_SWITCH_EN is shown in Table 5-68.

Return to the Summary Table.

Table 5-68. PSCON_RAM_FORCE_SWITCH_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	mem_all_switch_force_en	R/W	0h	When asserted all the VDDAR pg nets will be powered up state irrespective of the power domains state and PSCONs state 0x 0 -> VDDAR switches enable signal depends on PD sleep signal 0x 1 -> All VDDAR switches are turned on

5.2.1.66 PSCON_RAM_SWITCH_EN_OVERRIDE1 Register (Offset = 480h) [Reset = X]

PSCON_RAM_SWITCH_EN_OVERRIDE1 is shown in Table 5-69.

Return to the Summary Table.

Table 5-69. PSCON_RAM_SWITCH_EN_OVERRIDE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_app_pd_mem_grp2_switch_en	R/W	0h	Override control for the Power state of the VDDAR pg net in APP PD GROUP2. It can take VDDAR pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> APP PD GROUP2 VDDAR is powered down 0x 1 -> APP PD GROUP2 VDDAR is powered up
16	sel_ov_app_pd_mem_grp2_switch_en	R/W	0h	Mux select control for the Power state of the VDDAR pg net in APP PD GROUP2. It can take VDDAR pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value
15-2	RESERVED	R/W	X	
1	ov_app_pd_mem_grp1_switch_en	R/W	0h	Override control for the Power state of the VDDAR pg net in APP PD GROUP0x1. It can take VDDAR pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> APP PD GROUP0x1 VDDAR is powered down 0x 1 -> APP PD GROUP0x1 VDDAR is powered up
0	sel_ov_app_pd_mem_grp1_switch_en	R/W	0h	Mux select control for the Power state of the VDDAR pg net in APP PD GROUP0x1. It can take VDDAR pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value

5.2.1.67 PSCON_RAM_SWITCH_EN_OVERRIDE2 Register (Offset = 484h) [Reset = X]

PSCON_RAM_SWITCH_EN_OVERRIDE2 is shown in Table 5-70.

Return to the Summary Table.

Table 5-70. PSCON_RAM_SWITCH_EN_OVERRIDE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	

Table 5-70. PSCON_RAM_SWITCH_EN_OVERRIDE2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	ov_fec_pd_mem_grp4_switch_en	R/W	0h	Override control for the Power state of the VDDAR pg net in FEC PD. It can take VDDAR pg net to power down state or to power up state irrespective power state of the FEC PD. 0x 0 -> FEC PD VDDAR is powered down 0x 1 -> FEC PD VDDAR is powered up
16	sel_ov_fec_pd_mem_grp4_switch_en	R/W	0h	Mux select control for the Power state of the VDDAR pg net in FEC PD. It can take VDDAR pg net to power down state or to power up state irrespective power state of the FEC PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value
15-2	RESERVED	R/W	X	
1	ov_hwa_pd_mem_grp3_switch_en	R/W	0h	Override control for the Power state of the VDDAR pg net in HWA PD. It can take VDDAR pg net to power down state or to power up state irrespective power state of the HWA PD. 0x 0 -> HWA PD VDDAR is powered down 0x 1 -> HWA PD VDDAR is powered up
0	sel_ov_hwa_pd_mem_grp3_switch_en	R/W	0h	Mux select control for the Power state of the VDDAR pg net in HWA PD. It can take VDDAR pg net to power down state or to power up state irrespective power state of the HWA PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value

5.2.1.68 PSCOM_RAM_SWITCH_DELAY Register (Offset = 488h) [Reset = X]

PSCOM_RAM_SWITCH_DELAY is shown in Table 5-71.

Return to the Summary Table.

Table 5-71. PSCOM_RAM_SWITCH_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	X	
15-14	fec_pd_mem_grp4_switch_highres_lowres_delay	R/W	2h	FEC PD specific delay value between the highres and low res VDDAR switches
13-12	hwa_pd_mem_grp3_switch_highres_lowres_delay	R/W	2h	HWA PD specific delay value between the highres and low res VDDAR switches
11-10	app_pd_mem_grp2_switch_highres_lowres_delay	R/W	2h	APP PD group2 specific delay value between the highres and low res VDDAR switches
9-8	app_pd_mem_grp1_switch_highres_lowres_delay	R/W	2h	APP PD group0x1 specific delay value between the highres and low res VDDAR switches
7-3	RESERVED	R/W	X	
2-1	global_mem_switch_highres_lowres_delay	R/W	2h	Global delay value between the highres and low res VDDAR switches
0	use_mem_switch_delay_global	R/W	0h	It controls the source of the delay value between the highres and low res VDDAR switches 0x 0 -> selects the individual delay values for each VDDAR net 0x 1 -> selects the global_sram_switch_highres_lowres_delay_cfg delay value for all VDDAR net

5.2.1.69 PSCON_DFTRTA_OVERRIDE Register (Offset = 48Ch) [Reset = X]

PSCON_DFTRTA_OVERRIDE is shown in Table 5-72.

Return to the Summary Table.

Table 5-72. PSCON_DFTRTA_OVERRIDE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_dftrtagood	R/W	0h	Override for the dftrtagood signal state of all memories irrespective of the PSCONs state. 0x 0 -> All memories in non retention mode 0x 1 -> All memories in retention mode
16	sel_ov_dftrtagood	R/W	1h	Mux select control for the dftrtagood signal state of all memories irrespective of the PSCONs state. 0x 0 -> selects the functional value 0x 1 -> select the override value for all dftrtagood signals
15-2	RESERVED	R/W	X	
1	ov_dftrtaon	R/W	0h	Override for the dftrtaon signal state of all memories irrespective of the PSCONs state. 0x 0 -> All memories in non retention mode 0x 1 -> All memories in retention mode
0	sel_ov_dftrtaon	R/W	1h	Mux select control for the dftrtaon signal state of all memories irrespective of the PSCONs state. 0x 0 -> selects the functional value 0x 1 -> select the override value for all dftrtaon signals

5.2.1.70 PSCON_VNWA_SWITCH_EN1 Register (Offset = 490h) [Reset = X]

PSCON_VNWA_SWITCH_EN1 is shown in Table 5-73.

Return to the Summary Table.

Table 5-73. PSCON_VNWA_SWITCH_EN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_app_pd_mem_grp2_vnwa_switch_en	R/W	0h	Override for the Power state of the VNWA pg net in APP PD GROUP2. It can take VNWA pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> APP PD GROUP2 VNWA is powered down 0x 1 -> APP PD GROUP2 VNWA is powered up
16	sel_ov_app_pd_mem_grp2_vnwa_switch_en	R/W	0h	Mux select control for the Power state of the VNWA pg net in APP PD GROUP2. It can take VNWA pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value
15-10	RESERVED	R/W	X	
9	ov_app_pd_mem_grp1_vnwa_switch_en	R/W	0h	Override for the Power state of the VNWA pg net in APP PD GROUP0x1. It can take VNWA pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> APP PD GROUP0x1 VNWA is powered down 0x 1 -> APP PD GROUP0x1 VNWA is powered up

Table 5-73. PSCON_VNWA_SWITCH_EN1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	sel_ov_app_pd_mem_grp1_vnwa_switch_en	R/W	0h	Mux select control for the Power state of the VNWA pg net in APP PD GROUP0x1. It can take VNWA pg net to power down state or to power up state irrespective power state of the APP PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value
7-2	RESERVED	R/W	X	
1	ov_app_pd_mem_vnwa_switch_en	R/W	0h	Override for the Power state of the VNWA pg net in APP PD for memories without external switch. 0x 0 -> APP PD VNWA is powered down 0x 1 -> APP PD VNWA is powered up
0	sel_ov_app_pd_mem_vnwa_switch_en	R/W	0h	Mux select control for the Power state of the VNWA pg net in APP PD for memories without external switch. 0x 0 -> selects the functional value 0x 1 -> select the override value of value

5.2.1.71 PSCON_VNWA_SWITCH_EN2 Register (Offset = 494h) [Reset = X]

PSCON_VNWA_SWITCH_EN2 is shown in Table 5-74.

Return to the Summary Table.

Table 5-74. PSCON_VNWA_SWITCH_EN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_fec_pd_mem_grp4_vnwa_switch_en	R/W	0h	Override for the Power state of the VNWA pg net in FEC PD. It can take VNWA pg net to power down state or to power up state irrespective power state of the FEC PD. 0x 0 -> FEC PD VNWA is powered down 0x 1 -> FEC PD VNWA is powered up
16	sel_ov_fec_pd_mem_grp4_vnwa_switch_en	R/W	0h	Mux select control for the Power state of the VNWA pg net in FEC PD. It can take VNWA pg net to power down state or to power up state irrespective power state of the FEC PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value
15-10	RESERVED	R/W	X	
9	ov_fec_pd_mem_vnwa_switch_en	R/W	0h	Override for the Power state of the VNWA pg net in FEC PD for memories without external switch. 0x 0 -> FEC PD VNWA is powered down 0x 1 -> FEC PD VNWA is powered up
8	sel_ov_fec_pd_mem_vnwa_switch_en	R/W	0h	Mux select control for the Power state of the VNWA pg net in FEC PDfor memories without external switch. 0x 0 -> selects the functional value 0x 1 -> select the override value of value
7-2	RESERVED	R/W	X	
1	ov_hwa_pd_mem_grp3_vnwa_switch_en	R/W	0h	Override for the Power state of the VNWA pg net in HWA PD. It can take VNWA pg net to power down state or to power up state irrespective power state of the HWA PD. 0x 0 -> HWA PD VNWA is powered down 0x 1 -> HWA PD VNWA is powered up

Table 5-74. PSCON_VNWA_SWITCH_EN2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	sel_ov_hwa_pd_mem_grp3_vnwa_switch_en	R/W	0h	Mux select control for the Power state of the VNWA pg net in HWA PD. It can take VNWA pg net to power down state or to power up state irrespective power state of the HWA PD. 0x 0 -> selects the functional value 0x 1 -> select the override value of value

5.2.1.72 PSCON_SRAM_LDO_WEAK_PROCESS Register (Offset = 498h) [Reset = X]

PSCON_SRAM_LDO_WEAK_PROCESS is shown in Table 5-75.

Return to the Summary Table.

Table 5-75. PSCON_SRAM_LDO_WEAK_PROCESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	vnwa_switch_weak_process_at_deepsleep_exit	R/W	1h	State of the prcm_weak_process signal at deep sleep exit. 0x 0 -> prcm_weak_process is 0x0 at deep sleep exit. 0x 1 -> prcm_weak_process is 0x1 at deep sleep exit.
7-2	RESERVED	R/W	X	
1	ov_vnwa_switch_weak_process	R/W	0h	Override for the state of the prcm_weak_process signal. 0x 0 -> prcm_weak_process is 0x0 0x 1 -> prcm_weak_process is 0x1
0	sel_ov_vnwa_switch_weak_process	R/W	0h	Mux select control for the state of the prcm_weak_process signal. 0x 0 -> selects the efuse prcm_weak_process 0x 1 -> select the override value prcm_weak_process

5.2.1.73 CLKM_OSC_CLK_REQ Register (Offset = 49Ch) [Reset = X]

CLKM_OSC_CLK_REQ is shown in Table 5-76.

Return to the Summary Table.

Table 5-76. CLKM_OSC_CLK_REQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	ov_osc_clk_req	R/W	0h	Override for the Power state of the CLKM FSMs. It Can take the CLKM FSM forcefully to sleep state or to active state. 0x 0 -> CLKM FSM in sleep state 0x 1 -> CLKM FSM in active state
0	sel_ov_osc_clk_req	R/W	0h	Mux select control for the Power state of the CLKM FSMs. It Can take the CLKM FSM forcefully to sleep state or to active state. 0x 0 -> selects the functional OSC_CLK_REQ 0x 1 -> select the override value OSC_CLK_REQ

5.2.1.74 CLKM_OVERRIDE1 Register (Offset = 4A0h) [Reset = X]

CLKM_OVERRIDE1 is shown in Table 5-77.

Return to the Summary Table.

Table 5-77. CLKM_OVERRIDE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_slicer_bias_en	R/W	0h	Override for the slicer bias state irrespective of the CLKM FSM state. 0x 0 -> Slicer bias disable 0x 1 -> Slicer bias enable
16	sel_ov_slicer_bias_en	R/W	0h	Mux select control for the slicer bias state irrespective of the CLKM FSM state. 0x 0 -> selects the functional SLICER_BIAS_EN 0x 1 -> select the override value SLICER_BIAS_EN
15-2	RESERVED	R/W	X	
1	ov_slicer_ldo_en	R/W	0h	Override for the slicer LDO enable state irrespective of the CLKM FSM state. 0x 0 -> Slicer LDO disable 0x 1 -> Slicer LDO enable
0	sel_ov_slicer_ldo_en	R/W	0h	Mux select control for the slicer LDO enable state irrespective of the CLKM FSM state. 0x 0 -> selects the functional SLICER_LDO_EN 0x 1 -> select the override value SLICER_LDO_EN

5.2.1.75 CLKM_OVERRIDE2 Register (Offset = 4A4h) [Reset = X]

CLKM_OVERRIDE2 is shown in Table 5-78.

Return to the Summary Table.

Table 5-78. CLKM_OVERRIDE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_slicer_en	R/W	0h	Override for the slicer enable signal state irrespective of the CLKM FSM state. 0x 0 -> Slicer disable 0x 1 -> Slicer enable
16	sel_ov_slicer_en	R/W	0h	Mux select control for the slicer enable signal state irrespective of the CLKM FSM state. 0x 0 -> selects the functional SLICER_EN 0x 1 -> select the override value SLICER_EN
15-2	RESERVED	R/W	X	
1	ov_xtal_en	R/W	0h	Override for the XTAL oscillator enable signal state irrespective of the CLKM FSM state. 0x 0 -> XTAL oscillator enable 0x 1 -> XTAL oscillator diable
0	sel_ov_xtal_en	R/W	0h	Mux select control for the XTAL oscillator enable signal state irrespective of the CLKM FSM state. 0x 0 -> selects the functional XTAL_EN 0x 1 -> select the override value XTAL_EN

5.2.1.76 CLKM_OVERRIDE3 Register (Offset = 4A8h) [Reset = X]

CLKM_OVERRIDE3 is shown in Table 5-79.

Return to the Summary Table.

Table 5-79. CLKM_OVERRIDE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	ov_clkm_oscillator_clk_valid	R/W	0h	Override for the slicer enable signal state irrespective of the CLKM FSM state. 0x 0 -> OSC CLK valid disable 0x 1 -> OSC CLK valid enable
16	sel_ov_clkm_oscillator_clk_valid	R/W	0h	Mux select control for the slicer enable signal state irrespective of the CLKM FSM state. 0x 0 -> selects the functional OSC CLK valid 0x 1 -> select the override value OSC CLK valid
15-2	RESERVED	R/W	X	
1	ov_xtal_det_en	R/W	0h	Override for the XTAL detection enable signal state irrespective of the CLKM FSM state. 0x 0 -> XTAL detection disable 0x 1 -> XTAL detection enable
0	sel_ov_xtal_det_en	R/W	0h	Mux select control for the XTAL detection enable signal state irrespective of the CLKM FSM state. 0x 0 -> selects the functional XTAL_DET_EN 0x 1 -> select the override value XTAL_DET_EN

5.2.1.77 CLKM_OUTPUT_HOST_CLK_REQ_OVERRIDE Register (Offset = 4ACh) [Reset = X]

CLKM_OUTPUT_HOST_CLK_REQ_OVERRIDE is shown in Table 5-80.

Return to the Summary Table.

Table 5-80. CLKM_OUTPUT_HOST_CLK_REQ_OVERRIDE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	ov_host_clk_req	R/W	0h	Override for the host clock request 0x 0 -> No host clock request 0x 1 -> Host clock request
0	sel_ov_host_clk_req	R/W	0h	Mux select control for the host clock request 0x 0 -> selects the functional HOST_CLK_REQ 0x 1 -> select the override value HOST_CLK_REQ

5.2.1.78 CLKM_SEL_OV_XT_DRIVE Register (Offset = 4B0h) [Reset = X]

CLKM_SEL_OV_XT_DRIVE is shown in Table 5-81.

Return to the Summary Table.

Table 5-81. CLKM_SEL_OV_XT_DRIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	X	
21-17	ov_high_xt_drive	R/W	0h	Override for the high XT drive signal state
16	sel_ov_high_xt_drive	R/W	0h	Mux select control for the high XT drive signal state 0x 0 -> selects the functional low XT drive 0x 1 -> select the override low XT drive
15-6	RESERVED	R/W	X	
5-1	ov_low_xt_drive	R/W	0h	Override for the low XT drive signal state

Table 5-81. CLKM_SEL_OV_XT_DRIVE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	sel_ov_low_xt_drive	R/W	0h	Mux select control for the low XT drive signal state 0x 0 -> selects the functional low XT drive 0x 1 -> select the override low XT drive

5.2.1.79 CLKM_DELAY1 Register (Offset = 4B4h) [Reset = X]

CLKM_DELAY1 is shown in Table 5-82.

Return to the Summary Table.

Table 5-82. CLKM_DELAY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-16	clkm_extend_valid_delay	R/W	0h	This delay will extend the valid state of the oscillator clock after the clock request is deasserted.
15-5	RESERVED	R/W	X	
4-0	clkm_clk_req_delay	R/W	0h	It controls the delay between the clock request is assertion and the start of oscillator enable sequence.

5.2.1.80 CLKM_DELAY2 Register (Offset = 4B8h) [Reset = X]

CLKM_DELAY2 is shown in Table 5-83.

Return to the Summary Table.

Table 5-83. CLKM_DELAY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26-17	ov_clkm_slicer_en_delay	R/W	3h	Override for the slicer enable delay
16	sel_ov_clkm_slicer_en_delay	R/W	0h	Mux select control for the slicer enable delay 0x 0 -> selects the efuse value/default value 0x 1 -> select the override value
15-11	RESERVED	R/W	X	
10-1	ov_clkm_osc_en_delay	R/W	1Fh	Override for the oscillator enable delay
0	sel_ov_clkm_osc_en_delay	R/W	0h	Mux select control for the oscillator enable delay 0x 0 -> selects the efuse value/default value 0x 1 -> select the override value

5.2.1.81 CLKM_HOST_CLK_REQ_DELAY Register (Offset = 4BCh) [Reset = X]

CLKM_HOST_CLK_REQ_DELAY is shown in Table 5-84.

Return to the Summary Table.

Table 5-84. CLKM_HOST_CLK_REQ_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26-17	ov_clkm_host_clk_req_pulse_width	R/W	64h	Override for the host clock request pulse width
16	sel_ov_clkm_host_clk_req_pulse_width	R/W	0h	Mux select control for the host clock request pulse width 0x 0 -> selects the efuse value 0x 1 -> select the override value
15-11	RESERVED	R/W	X	
10-1	ov_clkm_host_clk_req_delay	R/W	1E0h	Override for the host clock request delay

Table 5-84. CLKM_HOST_CLK_REQ_DELAY Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	sel_ov_clkm_host_clk_req_delay	R/W	0h	Mux select control for the host clock request delay 0x 0 -> selects the efuse value 0x 1 -> select the override value

5.2.1.82 RST_OVERRIDE Register (Offset = 4C0h) [Reset = X]

RST_OVERRIDE is shown in Table 5-85.

Return to the Summary Table.

Table 5-85. RST_OVERRIDE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4	ov_pscon_por_rstn	R/W	1h	When asserted all the PSCON will be in reset state and the all the power control signals output of PSCONs will have reset state value. 0x 0 -> All PSCONs are in reset state 0x 1 -> PSCON in functional state
3	RESERVED	R/W	X	
2	ov_hwa_pd_por_rstn	R/W	1h	When asserted HWA PD will be in POR reset state. 0x 0 -> HWA PD in reset state 0x 1 -> HWA PD in functional state
1	ov_fec_pd_por_rstn	R/W	1h	When asserted FEC PD will be in POR reset state. 0x 0 -> FEC PD in reset state 0x 1 -> FEC PD in functional state
0	ov_app_pd_por_rstn	R/W	1h	When asserted APP PD will be in POR reset state. 0x 0 -> APP PD in reset state 0x 1 -> APP PD in functional state

5.2.1.83 RST_SOFT_RESET Register (Offset = 4C4h) [Reset = X]

RST_SOFT_RESET is shown in Table 5-86.

Return to the Summary Table.

Table 5-86. RST_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-16	warm_rstn_pulse_width	R/W	10h	Warm reset active low pulse width value.
15-1	RESERVED	R/W	X	
0	warm_reset_reqn	R/W	0h	Software warm reset request. Generates warm reset for entire device. Writing 1 to this bit will trigger the warm reset generation logic. This bit is self cleared bit.

5.2.1.84 RST_WDT_RESET_EN Register (Offset = 4C8h) [Reset = X]

RST_WDT_RESET_EN is shown in Table 5-87.

Return to the Summary Table.

Table 5-87. RST_WDT_RESET_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	

Table 5-87. RST_WDT_RESET_EN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	wd_reset_en	R/W	0h	Watchdog timer warm reset enable. 0x 0 -> watchdog req for warm reset disable 0x 1 -> watchdog req for warm reset enable

5.2.1.85 RST_APP_PD_SOFT_RESET Register (Offset = 4CCh) [Reset = X]

RST_APP_PD_SOFT_RESET is shown in Table 5-88.

Return to the Summary Table.

Table 5-88. RST_APP_PD_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-16	app_pd_warm_rstn_pulse_width	R/W	10h	APP PD Warm reset active low pulse width value.
15-1	RESERVED	R/W	X	
0	app_pd_warm_reset_reqn	R/W	0h	Software warm reset request. Generates warm reset for APP PD. Writing 1 to this bit will trigger the app pd warm reset generation logic. This bit is self cleared bit.

5.2.1.86 RST_FEC_PD_SOFT_RESET Register (Offset = 4D0h) [Reset = X]

RST_FEC_PD_SOFT_RESET is shown in Table 5-89.

Return to the Summary Table.

Table 5-89. RST_FEC_PD_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-16	fec_pd_warm_rstn_pulse_width	R/W	10h	FEC PD Warm reset active low pulse width value.
15-1	RESERVED	R/W	X	
0	fec_pd_warm_reset_reqn	R/W	0h	Software warm reset request. Generates warm reset for FEC PD. Writing 1 to this bit will trigger the fec pd warm reset generation logic. This bit is self cleared bit.

5.2.1.87 RST_HWA_PD_SOFT_RESET Register (Offset = 4D4h) [Reset = X]

RST_HWA_PD_SOFT_RESET is shown in Table 5-90.

Return to the Summary Table.

Table 5-90. RST_HWA_PD_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-16	hwa_pd_warm_rstn_pulse_width	R/W	10h	HWA PD Warm reset active low pulse width value.
15-1	RESERVED	R/W	X	

Table 5-90. RST_HWA_PD_SOFT_RESET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	hwa_pd_warm_reset_reqn	R/W	0h	Software warm reset request. Generates warm reset for HWA PD. Writing 1 to this bit will trigger the hwa pd warm reset generation logic. This bit is self cleared bit.

5.2.1.88 RST_SOFT_APP_CORE_SYSRESET_REQ Register (Offset = 4D8h) [Reset = X]

RST_SOFT_APP_CORE_SYSRESET_REQ is shown in Table 5-91.

Return to the Summary Table.

Table 5-91. RST_SOFT_APP_CORE_SYSRESET_REQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-16	app_pd_core_rstn_pulse_width	R/W	10h	APP PD Core reset active low pulse width value.
15-1	RESERVED	R/W	X	
0	app_pd_core_reset_reqn	R/W	0h	Software core sysreset request. Generates core sysreset for APP PD core. Writing 1 to this bit will trigger the app pd core reset generation logic. This bit is self cleared bit.

5.2.1.89 RST_SOFT_FEC_CORE_SYSRESET_REQ Register (Offset = 4DCh) [Reset = X]

RST_SOFT_FEC_CORE_SYSRESET_REQ is shown in Table 5-92.

Return to the Summary Table.

Table 5-92. RST_SOFT_FEC_CORE_SYSRESET_REQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20-16	fec_pd_core_rstn_pulse_width	R/W	10h	FEC PD Core reset active low pulse width value.
15-1	RESERVED	R/W	X	
0	fec_pd_core_reset_reqn	R/W	0h	Software core sysreset request. Generates core sysreset for FEC PD core. Writing 1 to this bit will trigger the fec pd core reset generation logic. This bit is self cleared bit.

5.2.1.90 SYS_RST_CAUSE Register (Offset = 4E0h) [Reset = X]

SYS_RST_CAUSE is shown in Table 5-93.

Return to the Summary Table.

Table 5-93. SYS_RST_CAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	sys_rst_cause_clr	R/W	0h	Clear's the sys_rst_cause register 0x 0 -> sys_rst_cause capture enable 0x 1 -> sys_rst_cause reg clear and disable
15-3	RESERVED	R/W	X	

Table 5-93. SYS_RST_CAUSE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	sys_rst_cause	R	0h	System Reset Cause register 3'b 001 - POR reset 3'b 010 - Warm reset due to soft register 3'b 100 - Warm reset due to wdog

5.2.1.91 DUBUGSS_DISABLE Register (Offset = 4E4h) [Reset = X]

DUBUGSS_DISABLE is shown in Table 5-94.

Return to the Summary Table.

Table 5-94. DUBUGSS_DISABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	debugss_disable	R/W	0h	When asserted it disables the debugss signals for the PRCM module. 0x 0 -> Allows debugss signals to change the states in the PRCM module 0x 1 -> Disable the debugss signals

5.2.1.92 SLOW_CLK_CLKCTL Register (Offset = 4E8h) [Reset = X]

SLOW_CLK_CLKCTL is shown in Table 5-95.

Return to the Summary Table.

Table 5-95. SLOW_CLK_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17-16	slow_clk_in_use	R	0h	Current Clock selected by GCM Clock Mux 01 -> RC oscillator clock as slow_clk 10 -> RTC clock as slow clock
15-3	RESERVED	R/W	X	
2-0	slow_clk_src_sel	R/W	0h	When asserted selects the RTC clock input as slow clock instead of the rc oscillator clock 0x 0 -> RC oscillator clock as slow_clk 0x 7 -> RTC clock as slow clock

5.2.1.93 DEBUGSS_CLK_CLKCTL Register (Offset = 4ECh) [Reset = X]

DEBUGSS_CLK_CLKCTL is shown in Table 5-96.

Return to the Summary Table.

Table 5-96. DEBUGSS_CLK_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	
23-16	debugss_clk_in_use	R	0h	Current Clock selected by GCM Clock Mux 0x 1 : SLOW_CLK 0x 2 : RCOSC10M 0x 4 : TOPSS_SYS_CLK
15-12	RESERVED	R/W	X	

Table 5-96. DEBUGSS_CLK_CLKCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-0	debugss_clk_src_sel	R/W	222h	Slect the source clock: 0x 0 : SLOW_CLK 0x 1 : RCOSC10M 0x 2 : TOPSS_SYS_CLK For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.

5.2.1.94 EFUSE_10M_OSC_DISABLE Register (Offset = 4F0h) [Reset = X]

EFUSE_10M_OSC_DISABLE is shown in Table 5-97.

Return to the Summary Table.

Table 5-97. EFUSE_10M_OSC_DISABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	ov_efuse_10MHz_osc_disable	R/W	0h	Override for the Efuse 10MHz oscillator clock disable 0x 0 -> 10MHz osc enable 0x 1 -> 10MHz osc disable
2-0	sel_ov_efuse_10MHz_osc_disable	R/W	0h	Mux select control for the Efuse 10MHz oscillator clock disable 0x 0 -> Selects the efuse_autoload_done as disable signal 0x 7 -> select the override value

5.2.1.95 DEBUGSS_CLK_AUTOSWITCH Register (Offset = 4F4h) [Reset = X]

DEBUGSS_CLK_AUTOSWITCH is shown in Table 5-98.

Return to the Summary Table.

Table 5-98. DEBUGSS_CLK_AUTOSWITCH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1-0	debugss_clk_autoswitch_en	R/W	0h	Enable for the Auto switch of debugss clock source using the APPSS PD ISO , 0x 0 : Auto switch disable 0x 1 : Auto switch to RCOSC10M 0x 2 : Auto switch to SLOW_CLK

5.2.1.96 RADAR_SAFTY_ERROR_REG Register (Offset = 4F8h) [Reset = X]

RADAR_SAFTY_ERROR_REG is shown in Table 5-99.

Return to the Summary Table.

Table 5-99. RADAR_SAFTY_ERROR_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	safety_error_reg_clear	R/W	0h	Safety error register clear
16	safety_error_reg_wr_dis	R/W	0h	Safety error write disable
15	RESERVED	R/W	X	
14-9	pscon_mem_fsm_unknown_state_error_reg	R	0h	Memory PSCON fsm unknown state error
8-5	pscon_logic_fsm_unknown_state_error_reg	R	0h	Logic PSCON fsm unknown state error

Table 5-99. RADAR_SAFTY_ERROR_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	clkm_fsm_unknown_state_error_reg	R	0h	CLKM fsm unknown state error
3	bgap_fsm_unknown_state_error_reg	R	0h	BGAP fsm unknown state error
2	dig_ldo_fsm_unknown_state_error_reg	R	0h	DIG LDO fsm unknown state error
1	sram_ldo_fsm_unknown_state_error_reg	R	0h	SRAM LDO fsm unknown state error
0	radar_state_fsm_unknown_state_error_reg	R	0h	Radar fsm unknown state error

5.2.1.97 RELEASEFROMWIR_REG Register (Offset = 4FCh) [Reset = X]

RELEASEFROMWIR_REG is shown in Table 5-100.

Return to the Summary Table.

Table 5-100. RELEASEFROMWIR_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	releasefromwir_fec_core_rstn	R/W	0h	Release from wait ni reset for FEC CORE rstn
15-1	RESERVED	R/W	X	
0	releasefromwir_app_core_rstn	R/W	0h	Release from wait ni reset for APP CORE rstn

5.2.1.98 HWA_PD_MEM_SHARE_REG Register (Offset = 500h) [Reset = X]

HWA_PD_MEM_SHARE_REG is shown in Table 5-101.

Return to the Summary Table.

Table 5-101. HWA_PD_MEM_SHARE_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	X	
10-8	hwa_pd_mem_share_fecss_config	R/W	0h	writing '111' will extend the CM3 code memory to 96KB shared memory.
7-6	RESERVED	R/W	X	
5-0	hwa_pd_mem_share_appss_config	R/W	0h	bit 2: 0 : writing '111' will extend the CM4 code memory to 128KB shared memory. Bit 5: 3 : writing '111' will extend the CM4 code memory to 256KB shared memory.

5.2.1.99 FRC_OSC_CLK_GATE Register (Offset = 504h) [Reset = X]

FRC_OSC_CLK_GATE is shown in Table 5-102.

Return to the Summary Table.

Table 5-102. FRC_OSC_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	

Table 5-102. FRC_OSC_CLK_GATE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	clk_gate	R/W	0h	FRC oscillator clock gate 0x 0 : Enable the Clock 0x 7 : Gate the clock

5.2.1.100 MEMSWAP_REG Register (Offset = 508h) [Reset = X]

MEMSWAP_REG is shown in Table 5-103.

Return to the Summary Table.

Table 5-103. MEMSWAP_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	mmr_fecmemswap_lock	R/W	1h	
0	mmr_appmemswap_lock	R/W	1h	

5.2.1.101 LIMP_MODE_STATUS Register (Offset = 50Ch) [Reset = X]

LIMP_MODE_STATUS is shown in Table 5-104.

Return to the Summary Table.

Table 5-104. LIMP_MODE_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	X	
1	limp_mode_rcosc10m	R	0h	Limp mode status for RCOSC10M clock
0	limp_mode_xtal_clk	R	0h	Limp mode status for XTAL clock

5.2.1.102 RTI_CLOCK_GATE_SLEEP_STATE Register (Offset = 510h) [Reset = X]

RTI_CLOCK_GATE_SLEEP_STATE is shown in Table 5-105.

Return to the Summary Table.

Table 5-105. RTI_CLOCK_GATE_SLEEP_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	fecss_rti_clk_gate_in_sleep	R/W	0h	This register controls the clock gating of the XTAL clock connected to the RTI GCM in FECSS 0x 0 : XTAL clock wont be gated in the device sleep state 0x 1 : XTAL clock will be gated in the device sleep state
7-1	RESERVED	R/W	X	
0	appss_rti_wd_clk_gate_in_sleep	R/W	0h	This register controls the clock gating of the XTAL clock connected to the RTI and WD GCM in APPSS 0x 0 : XTAL clock wont be gated in the device sleep state 0x 1 : XTAL clock will be gated in the device sleep state

5.2.1.103 TOP_3318_LDO_EN_CTRL Register (Offset = 514h) [Reset = X]

TOP_3318_LDO_EN_CTRL is shown in Table 5-106.

Return to the Summary Table.

Table 5-106. TOP_3318_LDO_EN_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	X	
28-24	TOP_3318_LDO_VTRIM	R/W	0h	
23-21	RESERVED	R/W	X	
20-16	TOP_3318_KA_LDO_VTRIM	R/W	0h	
15-9	RESERVED	R/W	X	
8	TOP_3318_LDO_EN	R/W	0h	
7-1	RESERVED	R/W	X	
0	TOP_3318_KA_LDO_EN	R/W	0h	

5.2.1.104 RFANA_TOP_LDO_EN Register (Offset = 518h) [Reset = X]

RFANA_TOP_LDO_EN is shown in Table 5-107.

Return to the Summary Table.

Table 5-107. RFANA_TOP_LDO_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	X	
19-18	SPARE1	R/W	0h	SPARE 0x 0 = Functional Reset
17	LODIST_LDO_EN	R/W	0h	LODIST LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
16	TX_ANA_LDO_EN	R/W	0h	TX_ANA LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
15	SYNTH_SDM_LDO_EN	R/W	0h	SYNTH SDM LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
14	SYNTH_DIV_LDO_EN	R/W	0h	SYNTH DIV LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
13	SYNTH_VCO_LDO_EN	R/W	0h	SYNTH VCO LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
12	APLL_OBUF_LDO_EN	R/W	0h	APLL OBUF LDO EN <0> LDO Disbaled <1> LDO Enabled 0x 0 = Functional Reset
11	APLL_VCO_LDO_EN	R/W	0h	APLL VCO LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
10	APLL_CP_LDO_EN	R/W	0h	APLL CP LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
9-6	SPARE0	R/W	0h	SPARE 0x 0 = Functional Reset
5	RX3_IFA_LDO_EN	R/W	0h	RX3 IFA LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
4	RX3_ADC_DIG_LDO_EN	R/W	0h	RX3 ADC DIG LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
3	RX2_IFA_LDO_EN	R/W	0h	RX2 IFA LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
2	RX2_ADC_DIG_LDO_EN	R/W	0h	RX2 ADC DIG LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
1	RX1_IFA_LDO_EN	R/W	0h	RX1 IFA LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset
0	RX1_ADC_DIG_LDO_EN	R/W	0h	RX1 ADC DIG LDO EN <0> LDO Disabled <1> LDO Enabled 0x 0 = Functional Reset

5.2.1.105 RFANA_TOP_ISO_CTRL Register (Offset = 51Ch) [Reset = X]

RFANA_TOP_ISO_CTRL is shown in Table 5-108.

Return to the Summary Table.

Table 5-108. RFANA_TOP_ISO_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	X	
19-18	SPARE1	R/W	1h	SPARE 0x 1 = Functional Reset
17	LODIST_LDO_ISO	R/W	1h	LODIST LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
16	TX_ANA_LDO_ISO	R/W	1h	TX_ANA LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
15	SYNTH_SDM_LDO_ISO	R/W	1h	SYNTH SDM LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
14	SYNTH_DIV_LDO_ISO	R/W	1h	SYNTH DIV LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
13	SYNTH_VCO_LDO_ISO	R/W	1h	SYNTH VCO LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
12	APLL_OBUF_LDO_ISO	R/W	1h	APLL OBUF LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
11	APLL_VCO_LDO_ISO	R/W	1h	APLL VCO LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
10	APLL_CP_LDO_ISO	R/W	1h	APLL CP LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
9	MDLL_CLK_ISO	R/W	1h	MDLL Clock ISO <0> CLK_MDLL_200M_LOWV active <1> CLK_MDLL_200M_LOWV disabled 0x 1 = Functional Reset
8-6	SPARE0	R/W	1h	SPARE 0x 1 = Functional Reset
5	RX3_IFA_LDO_ISO	R/W	1h	RX3 IFA LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
4	RX3_ADC_DIG_LDO_ISO	R/W	1h	RX3 ADC DIG LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
3	RX2_IFA_LDO_ISO	R/W	1h	RX2 IFA LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
2	RX2_ADC_DIG_LDO_ISO	R/W	1h	RX2 ADC DIG LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
1	RX1_IFA_LDO_ISO	R/W	1h	RX1 IFA LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset
0	RX1_ADC_DIG_ISO	R/W	1h	RX1 ADC DIG LDO ISO <0> Outputs on LDO supply enabled <1> Outputs on LDO supply disabled 0x 1 = Functional Reset

5.2.1.106 CLK_CTRL_REG1_LDO_CLKTOP Register (Offset = 520h) [Reset = 00400710h]

CLK_CTRL_REG1_LDO_CLKTOP is shown in Table 5-109.

Return to the Summary Table.

Table 5-109. CLK_CTRL_REG1_LDO_CLKTOP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
23-20	BISTMUX_CTRL	R/W	4h	SLICER LDO BIST MUX CONTROL (ONE HOT) Analog MUX enables to BIST output port 0000 = HI-Z Output 0001 = VBG_0P9*10/ 9 =1.0 V 0010 = VDD18*0. 5 = 0.9V 0100 = VLDO Output * 0.6 1000 = Floating WARNING: Enabling more than one bit may damage the device 0x 4 = Functional Reset
19-16	TESTMUX_CTRL	R/W	0h	SLICER LDO TEST MUX CONTROL (ONE HOT) Analog MUX enables to test output port 0000 = HI-Z Output 0001 = 0.6 * VLDO_OUT 0010 = VDD18*0. 5 = 0.9V 0100 = VSSA 1000 = LDO Test Current (12.5uA) WARNING: Enabling more than one bit may damage the device 0x 0 = Functional Reset
15-13	TLOAD_CTRL	R/W	0h	SLICER LDO TLOAD CONTROL Value should be 0x0 during boot sequence to ensure stability while unloaded, then 0x1 to turn off all current loading after oscillator is enabled to reduce power and extend reliability. Iload=undefined*24mA+undefined*16mA+!undefined*8mA 0b 001 = no current load 0b 000 = 8mA load 0b 010 = 16mA load 0b 100 = 24mA load 0x 0 = Functional Reset
12	ENABLE_PMOS_PULLDOWN	R/W	0h	SLICER LDO PMOS PULL DOWN ENABLE 0 = Slicer LDO PMOS Pull Down disabled 1 = Slicer LDO PMOS Pull Down enabled 0x 0 = Functional Reset
11	SCPRT_IBIAS_CTRL	R/W	0h	SLICER LDO SHORT CKT PROTECTION IBIAS CONTROL 0 = Nominal short circuit bias with nominal short circuit current limit 1 = 2X Nominal short circuit bias with higher short circuit current limit 0x 0 = Functional Reset
10-8	LDO_BW_CTRL	R/W	7h	SLICER LDO BANDWIDTH CONTROL Control the bias current in the fast loop buffer of the SLICER LDO, in steps of 10uA 101 - 30uA 111 - 50uA (default) 010 - 100uA 0x 7 = Functional Reset
7	EN_BYPASS	R/W	0h	SLICER LDO BYPASS ENABLE 0 = Slicer LDO in normal mode 1 = Slicer LDO Bypassed with external voltage 0x 0 = Functional Reset
6	EN_SHRT_CKT	R/W	0h	SLICER LDO SHORT CKT PROTECTION ENABLE 0 = Slicer LDO Short Ckt Protection Disabled 1 = Slicer LDO Short Ckt Protection Enabled 0x 0= Functional Reset

Table 5-109. CLK_CTRL_REG1_LDO_CLKTOP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	EN_TEST_MODE	R/W	0h	SLICER LDO TEST MODE ENABLE 0 = Slicer LDO TEST MODE Disabled 1 = Slicer LDO TEST MODE Enabled 0x 0 = Functional Reset
4	ENZ_LOW_BW_CAP	R/W	1h	SLICER LDO LOW BW MODE ENABLE 0 = Slicer LDO Low BW mode Disabled 1 = Slicer LDO Low BW mode Enabled 0x 1 = Functional Reset
3-0	LDO_VOUT_CTRL	R/W	0h	SLICER LDO VOUT TRIM Trim the LDO output voltage, in steps of 25mV 0000 - 1.40V 0001 - 1.375V 0010 - 1.35V 0011 - 1.325V 0100 - 1.30V 0101 - 1.275V 0110 - 1.25V 0111 - 1.225V 1000 - 1.60V 1001 - 1.575V 1010 - 1.55V 1011 - 1.525V 1100 - 1.50V 1101 - 1.475V 1110 - 1.45V 1111 - 1.425V 0x 0 = Functional Reset

5.2.1.107 CLK_CTRL_REG1_XO_SLICER Register (Offset = 524h) [Reset = 00000000h]

CLK_CTRL_REG1_XO_SLICER is shown in Table 5-110.

Return to the Summary Table.

Table 5-110. CLK_CTRL_REG1_XO_SLICER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	SPARE2	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
14	SLICER_APLL_BYPASS_DRV	R/W	0h	Slicer APLL Bypass Drive This bit controls the drive strength of the APLL Bypass Slicer 0 = Low-power drive 1 = High-power drive 0x 0 = Functional Reset
13	SLICER_APLL_BYPASS	R/W	0h	Slicer APLL Bypass This bit enables a high-speed slicer connected to CLKM which can be used to drive a high-speed clock directly as the SYNTH reference clock. 0 = Normal operation (bypass slicer disabled) 1 = APLL Bypass Slicer Enabled 0x 0 = Functional Reset
12	SPARE1	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
11	SLICER_DCCPL_XO_SLICER	R/W	0h	Slicer DC-Coupled Mode 0 = Normal operation (AC-couple CLKP to internal slicer) 1 = DC-couple CLKP to internal slicer to CLKP 0x 0 = Functional Reset

Table 5-110. CLK_CTRL_REG1_XO_SLICER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SLICER_HIPWR_XO_SLICER	R/W	0h	Slicer High-power Mode This bit bypasses the input clock slicer current-starving/filtering circuitry to increase gain and reduce device phase-noise at the expense of power and reduced supply noise rejection. This permits the use of a high-speed external test clock (660MHz max). 0 = Normal operation (current-limiting present) 1 = High-power/high-speed test mode 0x 0 = Functional Reset
9	FASTCHARGEZ_BIAS_XO_SLICER	R/W	0h	Bias Fast-charge Enable (Active Low) This bit bypasses the RC filtering on the XOSC/SLICER Bias to permit more rapid power-up. 0 = Bias fast-charge 1 = Normal operation (filtering present) 0x 0 = Functional Reset
8-4	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
3-0	RTRIM_BIAS_XO_SLICER	R/W	0h	Crystal Oscillator and Slicer Bias RTrim Binary-weighted bias control 0x 0 = Functional Reset

5.2.1.108 TOP_LDO_3318_CTRL_REG0 Register (Offset = 528h) [Reset = 008B2040h]

TOP_LDO_3318_CTRL_REG0 is shown in Table 5-111.

Return to the Summary Table.

Table 5-111. TOP_LDO_3318_CTRL_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	BIST_CTRL	R/W	0h	3318 ldo bist control 0x 0 = Functional Reset
27	BIST_EN	R/W	0h	3318 ldo bist enable 0x 0 = Functional Reset
26-12	LDO_CTRL	R/W	8B2h	3318 ldo ctrl 0x8B 2 = Functional Reset
11	LDO_EN	R/W	0h	3318 ldo enable 0x 0 = Functional Reset
10-3	LDO_SPARE	R/W	8h	3318 ldo spare 0x 0 = Functional Reset
2	LDO_TEST_EN	R/W	0h	3318 ldo test enable 0x 0 = Functional Reset
1-0	SPARE0	R/W	0h	3318 ldo spare 0x 0 = Functional Reset

5.2.1.109 TOP_LDO_3318_CTRL_REG1 Register (Offset = 52Ch) [Reset = 00000000h]

TOP_LDO_3318_CTRL_REG1 is shown in Table 5-112.

Return to the Summary Table.

Table 5-112. TOP_LDO_3318_CTRL_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	KALDO_EN	R/W	0h	3318 KA LDO EN 0x 0 = Functional Reset
30-16	KALDO_CTRL	R/W	0h	3318 KA LDO CTRL 0x 0 = Functional Reset

Table 5-112. TOP_LDO_3318_CTRL_REG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-0	KALDO_SPARE	R/W	0h	3318 KA LDO spare 0x 0 = Functional Reset

5.2.1.110 TOP_LDO_DIG_CTRL_REG0 Register (Offset = 530h) [Reset = 00000563h]

TOP_LDO_DIG_CTRL_REG0 is shown in Table 5-113.

Return to the Summary Table.

Table 5-113. TOP_LDO_DIG_CTRL_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SPARE2	R/W	0h	SPARE 0x 0 = Functional reset (dependent on supply configuration) This is now driven by FSM through DIG_LDO_EN control
30	SPARE1	R/W	0h	SPARE 0x 0 = Functional Reset (dependent on supply configuration) This is driven by the FSM through DIG_KA_LDO_EN
29-26	DKALDO_RTRIM	R/W	0h	DIG KA RTRIM 0x 0 = Functional Reset
25-22	DKALDO_TRIM	R/W	0h	DIG KA SPARE 0x 0 = Functional Reset
21-16	SPARE0	R/W	0h	SPARE This control is now driven from DIG_KA_LDO_VTRIM< 5:0>
15-12	DLDO_BIST_CTRL	R/W	0h	DIG LDO BIST CTRL 0x 0 = Functional Reset
11	DLDO_BIST_EN	R/W	0h	DIG LDO BIST EN 0x 0 = Functional Reset
10	DLDO_SC_EN	R/W	1h	DIG LDO EN SC protection 0x 1 = Functional Reset
9-8	DLDO_BW_CTRL	R/W	1h	DIG LDO bandwidth control 0x 1 = Functional Reset
7-0	DIGLDO_TRIM	R/W	63h	DIGLDO SPARE 0x 63 = Functional Reset <7> = 0 unused <6> = 1 En high second stage current <5> = 1 En High first stage current <4>= 0 Input Filter bypassz < 3:0> = Input filter trim 0000 = Filter bypass 0001 = 25K 0011 = 50K 0111 = 75K 1111 = 100K

5.2.1.111 TOP_LDO_DIG_CTRL_REG1 Register (Offset = 534h) [Reset = A0004000h]

TOP_LDO_DIG_CTRL_REG1 is shown in Table 5-114.

Return to the Summary Table.

Table 5-114. TOP_LDO_DIG_CTRL_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DLDO_INRUSH_CTRL_EN	R/W	1h	DIG LDO inrush ctrl lowv 0x 1 = Functional Reset
30	DLDO_TEST_EN	R/W	0h	DIG LDO TEST enable 0x 0 = Functional Reset

Table 5-114. TOP_LDO_DIG_CTRL_REG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	DKALDO_RTRIM_EN	R/W	1h	DIGKA RTRIM EN 0x 1 = Functional Reset 0 - Trim taken from TOP_LDO_DIG_CTRL_REG0 25:22> 1 - Trim taken from TOP_LDO_DIG_CTRL_REG0 29:26>
28-21	DLDO_TMUX_CTRL	R/W	0h	DIG LDO tmux ctrl 0x 0 = Functional Reset TMUX CTRL (EN TMUX gates bit < 3:0>) 0 - VDDA 1 - VOUT_INT 2 - VSSA 3 - IBIAS 4 - vout_pad (gated by KA/dig en) 5 - Needs to be 1 to pass bits < 3:0> to test out 6 - DIG short circuit protection output (gated by DIG en) 7 - IBIAS KA (gated by KA/dig en) Bits < 3:0> is gated bit 5 and tmux_en 7 6 5 4 3 2 1 0 - - - - - - - - 0 0 1 0 0 0 0 1 - VDDA 0 0 1 0 0 0 1 0 - vout_int 0 0 1 0 0 1 0 0 - vssa 0 0 1 0 1 0 0 0 - Ibias dig ldo 0 0 0 1 0 0 0 0 - vout_pad 0 1 0 0 0 0 0 0 - dig sc out 1 0 0 0 0 0 0 0 - ibias ka ldo
20-15	SPARE1	R/W	0h	SPARE This control is now driven from DIG_LDO_VTRIM< 5:0> 0x 0 = Functional Reset
14	DLDO_SC_TRIM	R/W	1h	DIG LDO SC CT trim 0x 0 = Functional Reset 0 - 700mA short circuit limit 1 - 1A short circuit limit
13	DLDO_BYPASS	R/W	0h	DIG LDO BYPASS 0x 0 = Functional Reset
12-0	SPARE0	R/W	0h	spare 0x 0 = Functional Reset

5.2.1.112 TOP_LDO_SRAM_CTRL_REG0 Register (Offset = 538h) [Reset = 00000563h]

TOP_LDO_SRAM_CTRL_REG0 is shown in Table 5-115.

Return to the Summary Table.

Table 5-115. TOP_LDO_SRAM_CTRL_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SPARE3	R/W	0h	SPARE 0x 0 = Functional Reset
30	SPARE2	R/W	0h	SPARE 0x 0 = Functional Reset
29-26	SKALDO_RTRIM	R/W	0h	SRAM KA RTRIM 0x 0 = Functional Reset
25-22	SPARE1	R/W	0h	SRAM KA SPARE 0x 0 = Functional Reset
21-16	SPARE0	R/W	0h	SPARE 0x 0 = Functional Reset
15-12	SLDO_BIST_CTRL	R/W	0h	SRAM LDO BIST CTRL 0x 0 = Functional Reset

Table 5-115. TOP_LDO_SRAM_CTRL_REG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SLDO_BIST_EN	R/W	0h	SRAM LDO BIST EN 0x 0 = Functional Reset
10	SLDO_SC_EN	R/W	1h	SRAM LDO EN SC protection 0x 1 = Functional Reset
9-8	SLDO_BW_CTRL	R/W	1h	SRAM LDO bandwidth control 0x 0 = Functional Reset
7-0	SLDO_TRIM	R/W	63h	SPARE 0x 63 = Functional Reset <7> = 0 unused <6> = 1 En high second stage current <5> = 1 En High first stage current <4>= 0 Input Filter bypassz < 3:0> = Input filter trim 0000 = Filter bypass 0001 = 25K 0011 = 50K 0111 = 75K 1111 = 100K

5.2.1.113 TOP_LDO_SRAM_CTRL_REG1 Register (Offset = 53Ch) [Reset = X]

TOP_LDO_SRAM_CTRL_REG1 is shown in Table 5-116.

Return to the Summary Table.

Table 5-116. TOP_LDO_SRAM_CTRL_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SLDO_INRUSH_CTRL_EN	R/W	1h	SLDO inrush control 0x 1 = Functional Reset
30	SLDO_TEST_EN	R/W	0h	SLDO TEST EN 0x 0 = Functional Reset
29	SKALDO_RTRIM_EN	R/W	1h	SKALDO RTRIM EN 0x 0 = Functional Reset 0 - Trim taken from TOP_LDO_SRAM_CTRL_REG0<25:22> 1 - Trim taken from TOP_LDO_SRAM_CTRL_REG0<29:26>
28-21	SLDO_TMUX_CTRL	R/W	0h	DIG LDO tmux ctrl 0x 0 = Functional Reset TMUX CTRL (EN TMUX gates bit <3:0>) 0 - VDDA 1 - VOUT_INT 2 - VSSA 3 - IBIAS 4 - vout_pad (gated by KA/dig en) 5 - Needs to be 1 to pass bits <3:0> to test out 6 - DIG short circuit protection output (gated by DIG en) 7 - IBIAS KA (gated by KA/dig en) Bits <3:0> is gated bit 5 and tmux_en 7 6 5 4 3 2 1 0 - - - - - - - - 0 0 1 0 0 0 0 1 - VDDA 0 0 1 0 0 0 1 0 - vout_int 0 0 1 0 0 1 0 0 - vssa 0 0 1 0 1 0 0 0 - Ibias dig ldo 0 0 0 1 0 0 0 0 - vout_pad 0 1 0 0 0 0 0 0 - dig sc out 1 0 0 0 0 0 0 0 - ibias ka ldo
20-15	SPARE1	R/W	0h	SPARE 0x 0 = Functional Reset
14	SLDO_SC_TRIM	R/W	1h	SRAM LDO SC CT trim 0x 0 = Functional Reset 0 - 700mA short circuit limit 1 - 1A short circuit limit

Table 5-116. TOP_LDO_SRAM_CTRL_REG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	SLDO_BYPASS	R/W	0h	SRAM LDO BYPASS 0x 0 = Functional Reset
12-10	RESERVED	R/W	X	
9-0	SPARE0	R/W	0h	spare 0x 0 = Functional Reset

5.2.1.114 LOCK0_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0_KICK0 is shown in Table 5-117.

Return to the Summary Table.

- KICK0 component

Table 5-117. LOCK0_KICK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

5.2.1.115 LOCK0_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0_KICK1 is shown in Table 5-118.

Return to the Summary Table.

- KICK1 component

Table 5-118. LOCK0_KICK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

5.2.1.116 intr_raw_status Register (Offset = 1010h) [Reset = X]

intr_raw_status is shown in Table 5-119.

Return to the Summary Table.

Interrupt Raw Status/Set Register

Table 5-119. intr_raw_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

5.2.1.117 intr_enabled_status_clear Register (Offset = 1014h) [Reset = X]

intr_enabled_status_clear is shown in Table 5-120.

Return to the Summary Table.

Interrupt Enabled Status/Clear register

Table 5-120. intr_enabled_status_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

5.2.1.118 intr_enable Register (Offset = 1018h) [Reset = X]

intr_enable is shown in Table 5-121.

Return to the Summary Table.

Interrupt Enable register

Table 5-121. intr_enable Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

5.2.1.119 intr_enable_clear Register (Offset = 101Ch) [Reset = X]

intr_enable_clear is shown in Table 5-122.

Return to the Summary Table.

Interrupt Enable Clear register

Table 5-122. intr_enable_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

5.2.1.120 eoi Register (Offset = 1020h) [Reset = X]

eoi is shown in Table 5-123.

Return to the Summary Table.

EOI register

Table 5-123. eoi Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

5.2.1.121 fault_address Register (Offset = 1024h) [Reset = 00000000h]

fault_address is shown in Table 5-124.

Return to the Summary Table.

Fault Address register

Table 5-124. fault_address Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

5.2.1.122 fault_type_status Register (Offset = 1028h) [Reset = X]

fault_type_status is shown in Table 5-125.

Return to the Summary Table.

Fault Type Status register

Table 5-125. fault_type_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	X	
6	fault_ns	R	0h	Non-secure access.

Table 5-125. fault_type_status Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	fault_type	R	0h	Fault Type 10_ 0000 = Supervisor read fault - priv = 1 dir = 1 dtype != 1 01_ 0000 = Supervisor write fault - priv = 1 dir = 0 00_ 1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_ 0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_ 0010 = User write fault - priv = 0 dir = 0 00_ 0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_ 0000 = No fault

5.2.1.123 fault_attr_status Register (Offset = 102Ch) [Reset = 00000000h]

fault_attr_status is shown in Table 5-126.

Return to the Summary Table.

Fault Attribute Status register

Table 5-126. fault_attr_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

5.2.1.124 fault_clear Register (Offset = 1030h) [Reset = X]

fault_clear is shown in Table 5-127.

Return to the Summary Table.

Fault Clear register

Table 5-127. fault_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	X	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

5.2.1.125 DEBUG_LOGIC_PSCON_OVERRIDE_SEL Register (Offset = 1800h) [Reset = X]

DEBUG_LOGIC_PSCON_OVERRIDE_SEL is shown in Table 5-128.

Return to the Summary Table.

Table 5-128. DEBUG_LOGIC_PSCON_OVERRIDE_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	X	
11	sel_ov_test_dbg_pd_iso	R/W	0h	Override select for test_dbg_pd_iso
10	sel_ov_test_dbg_pd_pgoodin	R/W	0h	Override select for test_dbg_pd_pgoodin
9	sel_ov_test_dbg_pd_ponin	R/W	0h	Override select for test_dbg_pd_ponin
8	sel_ov_hwa_pd_iso	R/W	0h	Override select for hwa_pd_iso
7	sel_ov_hwa_pd_pgoodin	R/W	0h	Override select for hwa_pd_pgoodin
6	sel_ov_hwa_pd_ponin	R/W	0h	Override select for hwa_pd_ponin
5	sel_ov_fec_pd_iso	R/W	0h	Override select for fec_pd_iso

Table 5-128. DEBUG_LOGIC_PSCON_OVERRIDE_SEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	sel_ov_fec_pd_pgoodin	R/W	0h	Override select for fec_pd_pgoodin
3	sel_ov_fec_pd_ponin	R/W	0h	Override select for fec_pd_ponin
2	sel_ov_app_pd_iso	R/W	0h	Override select for app_pd_iso
1	sel_ov_app_pd_pgoodin	R/W	0h	Override select for app_pd_pgoodin
0	sel_ov_app_pd_ponin	R/W	0h	Override select for app_pd_ponin

5.2.1.126 DEBUG_LOGIC_PSCON_OVERRIDE_VAL Register (Offset = 1804h) [Reset = X]

DEBUG_LOGIC_PSCON_OVERRIDE_VAL is shown in Table 5-129.

Return to the Summary Table.

Table 5-129. DEBUG_LOGIC_PSCON_OVERRIDE_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	X	
11	ov_test_dbg_pd_iso	R/W	0h	Override value for test_dbg_pd_iso
10	ov_test_dbg_pd_pgoodin	R/W	0h	Override value for test_dbg_pd_pgoodin
9	ov_test_dbg_pd_ponin	R/W	0h	Override value for test_dbg_pd_ponin
8	ov_hwa_pd_iso	R/W	0h	Override value for hwa_pd_iso
7	ov_hwa_pd_pgoodin	R/W	0h	Override value for hwa_pd_pgoodin
6	ov_hwa_pd_ponin	R/W	0h	Override value for hwa_pd_ponin
5	ov_fec_pd_iso	R/W	0h	Override value for fec_pd_iso
4	ov_fec_pd_pgoodin	R/W	0h	Override value for fec_pd_pgoodin
3	ov_fec_pd_ponin	R/W	0h	Override value for fec_pd_ponin
2	ov_app_pd_iso	R/W	0h	Override value for app_pd_iso
1	ov_app_pd_pgoodin	R/W	0h	Override value for app_pd_pgoodin
0	ov_app_pd_ponin	R/W	0h	Override value for app_pd_ponin

5.2.1.127 DEBUG_MEM_PSCON_OVERRIDE_SEL_1 Register (Offset = 1808h) [Reset = X]

DEBUG_MEM_PSCON_OVERRIDE_SEL_1 is shown in Table 5-130.

Return to the Summary Table.

Table 5-130. DEBUG_MEM_PSCON_OVERRIDE_SEL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	
23	sel_ov_app_pd_mem_grp2_dftrtagood	R/W	0h	Override select for app_pd_mem_grp2_dftrtagood
22	sel_ov_app_pd_mem_grp2_dftrtaon	R/W	0h	Override select for app_pd_mem_grp2_dftrtaon
21-20	sel_ov_app_pd_mem_grp1_dftrtagood	R/W	0h	Override select for app_pd_mem_grp1_dftrtagood
19-18	sel_ov_app_pd_mem_grp1_dftrtaon	R/W	0h	Override select for app_pd_mem_grp1_dftrtaon
17-15	sel_ov_app_pd_mem_dftrtagood	R/W	0h	Override select for app_pd_mem_dftrtagood
14-12	sel_ov_app_pd_mem_dftrtaon	R/W	0h	Override select for app_pd_mem_dftrtaon
11	sel_ov_app_pd_mem_grp2_agoodin	R/W	0h	Override select for app_pd_mem_grp2_agoodin

Table 5-130. DEBUG_MEM_PSCON_OVERRIDE_SEL_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	sel_ov_app_pd_mem_grp2_aonin	R/W	0h	Override select for app_pd_mem_grp2_aonin
9-8	sel_ov_app_pd_mem_grp1_agoodin	R/W	0h	Override select for app_pd_mem_grp1_agoodin
7-6	sel_ov_app_pd_mem_grp1_aonin	R/W	0h	Override select for app_pd_mem_grp1_aonin
5-3	sel_ov_app_pd_mem_agoodin	R/W	0h	Override select for app_pd_mem_agoodin
2-0	sel_ov_app_pd_mem_aonin	R/W	0h	Override select for app_pd_mem_aonin

5.2.1.128 DEBUG_MEM_PSCON_OVERRIDE_SEL_2 Register (Offset = 180Ch) [Reset = X]

DEBUG_MEM_PSCON_OVERRIDE_SEL_2 is shown in Table 5-131.

Return to the Summary Table.

Table 5-131. DEBUG_MEM_PSCON_OVERRIDE_SEL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	
23-22	sel_ov_fec_pd_mem_grp4_dftrtagood	R/W	0h	Override select for fec_pd_mem_grp4_dftrtagood
21-20	sel_ov_fec_pd_mem_grp4_dftrtaon	R/W	0h	Override select for fec_pd_mem_grp4_dftrtaon
19	sel_ov_fec_pd_mem_dftrtagood	R/W	0h	Override select for fec_pd_mem_dftrtagood
18	sel_ov_fec_pd_mem_dftrtaon	R/W	0h	Override select for fec_pd_mem_dftrtaon
17-16	sel_ov_fec_pd_mem_grp4_agoodin	R/W	0h	Override select for fec_pd_mem_grp4_agoodin
15-14	sel_ov_fec_pd_mem_grp4_aonin	R/W	0h	Override select for fec_pd_mem_grp4_aonin
13	sel_ov_fec_pd_mem_agoodin	R/W	0h	Override select for fec_pd_mem_agoodin
12	sel_ov_fec_pd_mem_aonin	R/W	0h	Override select for fec_pd_mem_aonin
11-9	sel_ov_hwa_pd_mem_grp3_dftrtagood	R/W	0h	Override select for hwa_pd_mem_grp3_dftrtagood
8-6	sel_ov_hwa_pd_mem_grp3_dftrtaon	R/W	0h	Override select for hwa_pd_mem_grp3_dftrtaon
5-3	sel_ov_hwa_pd_mem_grp3_agoodin	R/W	0h	Override select for hwa_pd_mem_grp3_agoodin
2-0	sel_ov_hwa_pd_mem_grp3_aonin	R/W	0h	Override select for hwa_pd_mem_grp3_aonin

5.2.1.129 DEBUG_MEM_PSCON_OVERRIDE_VAL_1 Register (Offset = 1810h) [Reset = X]

DEBUG_MEM_PSCON_OVERRIDE_VAL_1 is shown in Table 5-132.

Return to the Summary Table.

Table 5-132. DEBUG_MEM_PSCON_OVERRIDE_VAL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	

Table 5-132. DEBUG_MEM_PSCON_OVERRIDE_VAL_1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	ov_app_pd_mem_grp2_dftrtagood	R/W	0h	Override value for app_pd_mem_grp2_dftrtagood
22	ov_app_pd_mem_grp2_dftrtaon	R/W	0h	Override value for app_pd_mem_grp2_dftrtaon
21-20	ov_app_pd_mem_grp1_dftrtagood	R/W	0h	Override value for app_pd_mem_grp1_dftrtagood
19-18	ov_app_pd_mem_grp1_dftrtaon	R/W	0h	Override value for app_pd_mem_grp1_dftrtaon
17-15	ov_app_pd_mem_dftrtagood	R/W	0h	Override value for app_pd_mem_dftrtagood
14-12	ov_app_pd_mem_dftrtaon	R/W	0h	Override value for app_pd_mem_dftrtaon
11	ov_app_pd_mem_grp2_agoodin	R/W	0h	Override value for app_pd_mem_grp2_agoodin
10	ov_app_pd_mem_grp2_aonin	R/W	0h	Override value for app_pd_mem_grp2_aonin
9-8	ov_app_pd_mem_grp1_agoodin	R/W	0h	Override value for app_pd_mem_grp1_agoodin
7-6	ov_app_pd_mem_grp1_aonin	R/W	0h	Override value for app_pd_mem_grp1_aonin
5-3	ov_app_pd_mem_agoodin	R/W	0h	Override value for app_pd_mem_agoodin
2-0	ov_app_pd_mem_aonin	R/W	0h	Override value for app_pd_mem_aonin

5.2.1.130 DEBUG_MEM_PSCON_OVERRIDE_VAL_2 Register (Offset = 1814h) [Reset = X]

DEBUG_MEM_PSCON_OVERRIDE_VAL_2 is shown in Table 5-133.

Return to the Summary Table.

Table 5-133. DEBUG_MEM_PSCON_OVERRIDE_VAL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	
23-22	ov_fec_pd_mem_grp4_dftrtagood	R/W	0h	Override value for fec_pd_mem_grp4_dftrtagood
21-20	ov_fec_pd_mem_grp4_dftrtaon	R/W	0h	Override value for fec_pd_mem_grp4_dftrtaon
19	ov_fec_pd_mem_dftrtagood	R/W	0h	Override value for fec_pd_mem_dftrtagood
18	ov_fec_pd_mem_dftrtaon	R/W	0h	Override value for fec_pd_mem_dftrtaon
17-16	ov_fec_pd_mem_grp4_agoodin	R/W	0h	Override value for fec_pd_mem_grp4_agoodin
15-14	ov_fec_pd_mem_grp4_aonin	R/W	0h	Override value for fec_pd_mem_grp4_aonin
13	ov_fec_pd_mem_agoodin	R/W	0h	Override value for fec_pd_mem_agoodin
12	ov_fec_pd_mem_aonin	R/W	0h	Override value for fec_pd_mem_aonin
11-9	ov_hwa_pd_mem_grp3_dftrtagood	R/W	0h	Override value for hwa_pd_mem_grp3_dftrtagood
8-6	ov_hwa_pd_mem_grp3_dftrtaon	R/W	0h	Override value for hwa_pd_mem_grp3_dftrtaon
5-3	ov_hwa_pd_mem_grp3_agoodin	R/W	0h	Override value for hwa_pd_mem_grp3_agoodin
2-0	ov_hwa_pd_mem_grp3_aonin	R/W	0h	Override value for hwa_pd_mem_grp3_aonin

5.2.1.131 MCUCLKOUT_CLKCTL Register (Offset = 1C00h) [Reset = X]

MCUCLKOUT_CLKCTL is shown in Table 5-134.

Return to the Summary Table.

Table 5-134. MCUCLKOUT_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	mcuclkout_clk_divr	R/W	0h	Divide value 0x 0 : div1 0x 1 : div2 0x 2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	mcuclkout_clk_src_sel	R/W	0h	Slect the source clock: 0x 0 : XTALCLK 0x 1 : MDLL 0x 2 : APLL/DPLL For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.
3-0	mcuclkout_clk_sw_gate	R/W	7h	0x 0 : Enable the Clock 0x 7 : Gate the clock

5.2.1.132 MCUCLKOUT_CLKSTAT Register (Offset = 1C04h) [Reset = X]

MCUCLKOUT_CLKSTAT is shown in Table 5-135.

Return to the Summary Table.

Table 5-135. MCUCLKOUT_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	mcuclkout_clk_in_use	R	0h	Current Clock selected by GCM Clock Mux 0x 0 : XTALCLK 0x 1 : MDLL 0x 2 : APLL/DPLL
3-0	mcuclkout_clk_curr_divr	R	0h	Gives the current divr setting used by the clock divider.

5.2.1.133 DCDC_CTRL_REG1 Register (Offset = 1C08h) [Reset = X]

DCDC_CTRL_REG1 is shown in Table 5-136.

Return to the Summary Table.

Table 5-136. DCDC_CTRL_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	dcdc_clk_en	R/W	0h	PMIC Clockout DCDC Clock Enable
15-1	RESERVED	R/W	X	
0	dcdc_rstn_reg	R/W	0h	Reset control for PMIC DCDC 0x 0 -> Reset is not asserted by SW 0x 1 -> Reset is asserted by SW

5.2.1.134 DCDC_CTRL_REG2 Register (Offset = 1C0Ch) [Reset = X]

DCDC_CTRL_REG2 is shown in Table 5-137.

Return to the Summary Table.

Table 5-137. DCDC_CTRL_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	dcdc_freq_acc_mode	R/W	0h	PMIC Clockout DCDC Freq Acc Enable
15-1	RESERVED	R/W	X	
0	dcdc_dither_en	R/W	0h	PMIC Clockout DCDC Clock Dither Enable

5.2.1.135 DCDC_CTRL_REG3 Register (Offset = 1C10h) [Reset = X]

DCDC_CTRL_REG3 is shown in Table 5-138.

Return to the Summary Table.

Table 5-138. DCDC_CTRL_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	
23-16	dcdc_max_freq_thr	R/W	0h	PMIC Clockout DCDC Maximum Frequency Threshold
15-8	RESERVED	R/W	X	
7-0	dcdc_min_freq_thr	R/W	0h	PMIC Clockout DCDC Minimum Frequency Threshold

5.2.1.136 DCDC_SLOPE_REG Register (Offset = 1C14h) [Reset = X]

DCDC_SLOPE_REG is shown in Table 5-139.

Return to the Summary Table.

Table 5-139. DCDC_SLOPE_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26-0	dcdc_slope_val	R/W	0h	PMIC Clockout DCDC Slope Config Value

5.2.1.137 APP_CPU_CLKCTL Register (Offset = 1C18h) [Reset = X]

APP_CPU_CLKCTL is shown in Table 5-140.

Return to the Summary Table.

Table 5-140. APP_CPU_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-0	gate	R/W	0h	APP CPU Clock Enable 0x 0 : Enable the Clock 0x 7 : Gate the clock

5.2.1.138 VNWA_SWITCH_SCREEN_ENABLE Register (Offset = 1C1Ch) [Reset = X]

VNWA_SWITCH_SCREEN_ENABLE is shown in Table 5-141.

Return to the Summary Table.

Table 5-141. VNWA_SWITCH_SCREEN_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	enable	R/W	0h	Writing 1'b 1 : Enable the screen mode for VNWA, it selects the VNWA PAD source for memories 1'b 0 : Disable the screen mode for VNWA

5.2.1.139 PLLDIG_RST_SRC_SEL Register (Offset = 1C20h) [Reset = X]

PLLDIG_RST_SRC_SEL is shown in Table 5-142.

Return to the Summary Table.

Table 5-142. PLLDIG_RST_SRC_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	sel	R/W	0h	Writing 1'b 1 : Change the reset source for PLLDIG to only POR reset 1'b 0 : Reset source is the TOPSS sys rstn

5.2.1.140 MEM_POWERDOWN_ACCESS_ERR_DIS Register (Offset = 1C24h) [Reset = X]

MEM_POWERDOWN_ACCESS_ERR_DIS is shown in Table 5-143.

Return to the Summary Table.

Table 5-143. MEM_POWERDOWN_ACCESS_ERR_DIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	disable	R/W	0h	Writing 1'b 1 : disable the bus access error generation when memory is power down 1'b 0 : Access errors will generate when memories are power down

5.2.1.141 MEM_SWAP Register (Offset = 1C28h) [Reset = X]

MEM_SWAP is shown in Table 5-144.

Return to the Summary Table.

Table 5-144. MEM_SWAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	X	
1	APP_CPU_ECLIPSE_STATUS	R	0h	APP cpu eclipse status
0	FEC_CPU_ECLIPSE_STATUS	R	0h	FEC cpu eclipse status

5.2.1.142 SPARE_REG Register (Offset = 1C2Ch) [Reset = X]

SPARE_REG is shown in Table 5-145.

Return to the Summary Table.

Table 5-145. SPARE_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	

Table 5-145. SPARE_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-16	read_only	R	0h	SPARE registers read only
15-8	RESERVED	R/W	X	
7-0	read_write	R/W	0h	SPARE registers read-write

5.2.2 APP_RCM Registers

Table 5-146 lists the memory-mapped registers for the APP_RCM registers. All register offset addresses not listed in Table 5-146 should be considered as reserved locations and the register contents should not be modified.

Table 5-146. APP_RCM Registers

Offset	Acronym	Register Name	Section
0h	PID	PID register	Go
4h	APP_CPU_CLKCTL		Go
8h	APP_CPU_CLKSTAT		Go
Ch	APP_CAN_CLKCTL		Go
10h	APP_CAN_CLKSTAT		Go
14h	APP_SPI_CLKCTL		Go
18h	APP_SPI_CLKSTAT		Go
1Ch	APP_SPI_BUSIF_CLKCTL		Go
20h	APP_SPI_BUSIF_CLKSTAT		Go
24h	APP_QSPI_CLKCTL		Go
28h	APP_QSPI_CLKSTAT		Go
2Ch	TOPSS_CLKCTL		Go
30h	TOPSS_CLKSTAT		Go
34h	APP_RTI_CLKCTL		Go
38h	APP_RTI_CLKSTAT		Go
3Ch	APP_WD_CLKCTL		Go
40h	APP_WD_CLKSTAT		Go
44h	APP_UART_0_CLKCTL		Go
48h	APP_UART_0_CLKSTAT		Go
4Ch	APP_UART_1_CLKCTL		Go
50h	APP_UART_1_CLKSTAT		Go
54h	APP_I2C_CLKCTL		Go
58h	APP_I2C_CLKSTAT		Go
5Ch	APP_LIN_CLKCTL		Go
60h	APP_LIN_CLKSTAT		Go
64h	RESERVED0		Go
68h	RESERVED1		Go
6Ch	RESERVED2		Go
70h	RESERVED3		Go
74h	IPCFGCLKGATE0		Go
78h	IPCFGCLKGATE1		Go
7Ch	IPCFGCLKGATE2		Go
80h	BLOCKRESET0		Go
84h	BLOCKRESET1		Go
88h	BLOCKRESET2		Go
8Ch	PLATFORM_SIGNATURE		Go
90h	POWERMODE		Go
94h	RST_WFICHECK		Go
98h	RST_ASSERTDLY		Go
9Ch	RST2ASSERTDLY		Go
A0h	RST_FSM_TRIG		Go
A4h	RST_CAUSE		Go

Table 5-146. APP_RCM Registers (continued)

Offset	Acronym	Register Name	Section
A8h	RST_CAUSE_CLR		Go
ACh	XTALCLK_CLK_GATE		Go
B0h	XTALCLKX2_CLK_GATE		Go
B4h	APLLDIV2_CLK_GATE		Go
B8h	DFT_APPSS_LSTC_CLK_GATE		Go
BCh	DFT_APPSS_LSTC_VBUSP_CLK_GATE		Go
C0h	APP_ROM_CLOCK_GATE		Go
C4h	APP_RAM1_CLOCK_GATE		Go
C8h	APP_RAM2_CLOCK_GATE		Go
CCh	APP_RAM3_CLOCK_GATE		Go
D0h	CFG_XBARA_DYNAMIC_CG		Go
D4h	CFG_TPTC1_DYNAMIC_CG		Go
D8h	CFG_TPTC2_DYNAMIC_CG		Go
DCh	CFG_XBARA_SET_DYNAMIC_CG		Go
E0h	CFG_TPTC1_SET_DYNAMIC_CG		Go
E4h	CFG_TPTC2_SET_DYNAMIC_CG		Go
E8h	CM4_FORCE_HCLK_GATE		Go
ECh	LIN_SCI_DIV		Go
F0h	APP_LSTC_EN		Go
1008h	LOCK0_KICK0	- KICK0 component	Go
100Ch	LOCK0_KICK1	- KICK1 component	Go
1010h	INTR_RAW_STATUS	Interrupt Raw Status/Set Register	Go
1014h	INTR_ENABLED_STATUS_CLEAR	Interrupt Enabled Status/Clear register	Go
1018h	INTR_ENABLE	Interrupt Enable register	Go
101Ch	INTR_ENABLE_CLEAR	Interrupt Enable Clear register	Go
1020h	EOI	EOI register	Go
1024h	FAULT_ADDRESS	Fault Address register	Go
1028h	FAULT_TYPE_STATUS	Fault Type Status register	Go
102Ch	FAULT_ATTR_STATUS	Fault Attribute Status register	Go
1030h	FAULT_CLEAR	Fault Clear register	Go

Complex bit access types are encoded to fit into small table cells. Table 5-147 shows the codes that are used for access types in this section.

Table 5-147. APP_RCM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

5.2.2.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in Table 5-148.

Return to the Summary Table.

PID register

Table 5-148. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PID_MSB16	R	6180h	
15-11	PID_MISC	R	0h	
10-8	PID_MAJOR	R	2h	
7-6	PID_CUSTOM	R	0h	
5-0	PID_MINOR	R	14h	

5.2.2.2 APP_CPU_CLKCTL Register (Offset = 4h) [Reset = X]

APP_CPU_CLKCTL is shown in Table 5-149.

Return to the Summary Table.

Table 5-149. APP_CPU_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	SRCSEL	R/W	0h	Select the source clock: 0x0 : OSC_CLK 0x1 : SLOW_CLK 0x2 : MDLL_CLK 0x3 : FAST_CLK 0x4 : SLOW_CLK 0x5 : SLOW_CLK 0x6 : SLOW_CLK 0x7 : SLOW_CLK For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.
3-0	RESERVED	R/W	0h	Reserved

5.2.2.3 APP_CPU_CLKSTAT Register (Offset = 8h) [Reset = X]

APP_CPU_CLKSTAT is shown in Table 5-150.

Return to the Summary Table.

Table 5-150. APP_CPU_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	CURRCLK	R	1h	Current Clock selected by GCM Clock Mux 0x1 : XTALCLK 0x2 : XTALCLKX2 0x4 : MDLL 0x8 : APLL/DPLL 0x10 : RCCLK

Table 5-150. APP_CPU_CLKSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.4 APP_CAN_CLKCTL Register (Offset = Ch) [Reset = X]

APP_CAN_CLKCTL is shown in Table 5-151.

Return to the Summary Table.

Table 5-151. APP_CAN_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	SRCSEL	R/W	0h	0x0 : OSC_CLK 0x1 : OSC_CLKX2 0x2 : MDLL_CLK 0x3 : FAST_CLK 0x4 : SLOW_CLK 0x5 : SLOW_CLK 0x6 : SLOW_CLK 0x7 : SLOW_CLKFor other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.5 APP_CAN_CLKSTAT Register (Offset = 10h) [Reset = X]

APP_CAN_CLKSTAT is shown in Table 5-152.

Return to the Summary Table.

Table 5-152. APP_CAN_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	CURRCLK	R	0h	Current Clock selected by GCM Clock Mux 0x1 : XTALCLK 0x2 : XTALCLKX2 0x4 : MDLL 0x8 : APLL/DPLL 0x10 : RCCLK
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.6 APP_SPI_CLKCTL Register (Offset = 14h) [Reset = X]

APP_SPI_CLKCTL is shown in Table 5-153.

Return to the Summary Table.

Table 5-153. APP_SPI_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	

Table 5-153. APP_SPI_CLKCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	SRCSEL	R/W	0h	0x0 : OSC_CLK 0x1 : OSC_CLKX2 0x2 : MDLL_CLK 0x3 : FAST_CLK 0x4 : SLOW_CLK 0x5 : SLOW_CLK 0x6 : SLOW_CLK 0x7 : SLOW_CLK For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.7 APP_SPI_CLKSTAT Register (Offset = 18h) [Reset = X]

APP_SPI_CLKSTAT is shown in Table 5-154.

Return to the Summary Table.

Table 5-154. APP_SPI_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	CURRCLK	R	0h	Current Clock selected by GCM Clock Mux 0x1 : XTALCLK 0x2 : XTALCLKX2 0x4 : MDLL 0x8 : APLL/DPLL 0x10 : RCCLK
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.8 APP_SPI_BUSIF_CLKCTL Register (Offset = 1Ch) [Reset = X]

APP_SPI_BUSIF_CLKCTL is shown in Table 5-155.

Return to the Summary Table.

Table 5-155. APP_SPI_BUSIF_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	X	
11-0	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.

5.2.2.9 APP_SPI_BUSIF_CLKSTAT Register (Offset = 20h) [Reset = X]

APP_SPI_BUSIF_CLKSTAT is shown in Table 5-156.

Return to the Summary Table.

Table 5-156. APP_SPI_BUSIF_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.10 APP_QSPI_CLKCTL Register (Offset = 24h) [Reset = X]

APP_QSPI_CLKCTL is shown in Table 5-157.

Return to the Summary Table.

Table 5-157. APP_QSPI_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	SRCSEL	R/W	0h	0x0 : OSC_CLK 0x1 : OSC_CLKX2 0x2 : MDLL_CLK 0x3 : FAST_CLK 0x4 : SLOW_CLK 0x5 : SLOW_CLK 0x6 : SLOW_CLK 0x7 : SLOW_CLK For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.11 APP_QSPI_CLKSTAT Register (Offset = 28h) [Reset = X]

APP_QSPI_CLKSTAT is shown in Table 5-158.

Return to the Summary Table.

Table 5-158. APP_QSPI_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	CURRCLK	R	0h	Current Clock selected by GCM Clock Mux 0x1 : XTALCLK 0x2 : XTALCLKX2 0x4 : MDLL 0x8 : APLL/DPLL 0x10 : RCCLK
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.12 TOPSS_CLKCTL Register (Offset = 2Ch) [Reset = X]

TOPSS_CLKCTL is shown in Table 5-159.

Return to the Summary Table.

Table 5-159. TOPSS_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	SRCSEL	R/W	0h	0x0 : OSC_CLK 0x1 : SLOW_CLK 0x2 : MDLL 0x3 : FAST_CLK 0x4 : SLOW_CLK For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.
3-0	GATE	R/W	0h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.13 TOPSS_CLKSTAT Register (Offset = 30h) [Reset = X]

TOPSS_CLKSTAT is shown in Table 5-160.

Return to the Summary Table.

Table 5-160. TOPSS_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	CURRCLK	R	1h	Current Clock selected by GCM Clock Mux 0x1 : XTALCLK 0x2 : XTALCLKX2 0x4 : MDLL 0x8 : APLL/DPLL 0x10 : RCCLK
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.14 APP_RTI_CLKCTL Register (Offset = 34h) [Reset = X]

APP_RTI_CLKCTL is shown in Table 5-161.

Return to the Summary Table.

Table 5-161. APP_RTI_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.

Table 5-161. APP_RTI_CLKCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-4	SRCSEL	R/W	0h	0x0 : OSC_CLK 0x1 : XREF_IN 0x2 : OSC_CLK (Ungated OSC_CLK for RTI in Sleep mode) 0x3 : SLOW_CLK 0x4 : SLOW_CLK 0x5 : SLOW_CLK 0x6 : SLOW_CLK 0x7 : SLOW_CLK For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.15 APP_RTI_CLKSTAT Register (Offset = 38h) [Reset = X]

APP_RTI_CLKSTAT is shown in Table 5-162.

Return to the Summary Table.

Table 5-162. APP_RTI_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	CURRCLK	R	0h	Current Clock selected by GCM Clock Mux 0x1 : XTALCLK 0x2 : XTALCLKX2 0x4 : MDLL 0x8 : APLL/DPLL 0x10 : RCCLK
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.16 APP_WD_CLKCTL Register (Offset = 3Ch) [Reset = X]

APP_WD_CLKCTL is shown in Table 5-163.

Return to the Summary Table.

Table 5-163. APP_WD_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	SRCSEL	R/W	0h	0x0 : OSC_CLK 0x1 : XREF_IN 0x2 : OSC_CLK 0x3 : SLOW_CLK 0x4 : SLOW_CLK 0x5 : SLOW_CLK 0x6 : SLOW_CLK 0x7 : SLOW_CLK For other values if the lower 3 bits matches with above, corresponding clock is selected. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register.

Table 5-163. APP_WD_CLKCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.17 APP_WD_CLKSTAT Register (Offset = 40h) [Reset = X]

APP_WD_CLKSTAT is shown in Table 5-164.

Return to the Summary Table.

Table 5-164. APP_WD_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	X	
11-4	CURRCLK	R	0h	Current Clock selected by GCM Clock Mux 0x1 : XTALCLK 0x2 : XTALCLKX2 0x4 : MDLL 0x8 : APLL/DPLL 0x10 : RCCLK
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.18 APP_UART_0_CLKCTL Register (Offset = 44h) [Reset = X]

APP_UART_0_CLKCTL is shown in Table 5-165.

Return to the Summary Table.

Table 5-165. APP_UART_0_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	RESERVED	R/W	X	
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.19 APP_UART_0_CLKSTAT Register (Offset = 48h) [Reset = X]

APP_UART_0_CLKSTAT is shown in Table 5-166.

Return to the Summary Table.

Table 5-166. APP_UART_0_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.20 APP_UART_1_CLKCTL Register (Offset = 4Ch) [Reset = X]

APP_UART_1_CLKCTL is shown in Table 5-167.

Return to the Summary Table.

Table 5-167. APP_UART_1_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	RESERVED	R/W	X	
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.21 APP_UART_1_CLKSTAT Register (Offset = 50h) [Reset = X]

APP_UART_1_CLKSTAT is shown in Table 5-168.

Return to the Summary Table.

Table 5-168. APP_UART_1_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.22 APP_I2C_CLKCTL Register (Offset = 54h) [Reset = X]

APP_I2C_CLKCTL is shown in Table 5-169.

Return to the Summary Table.

Table 5-169. APP_I2C_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	RESERVED	R/W	X	
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.23 APP_I2C_CLKSTAT Register (Offset = 58h) [Reset = X]

APP_I2C_CLKSTAT is shown in Table 5-170.

Return to the Summary Table.

Table 5-170. APP_I2C_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.24 APP_LIN_CLKCTL Register (Offset = 5Ch) [Reset = X]

APP_LIN_CLKCTL is shown in Table 5-171.

Return to the Summary Table.

Table 5-171. APP_LIN_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-16	DIVR	R/W	0h	Divide value 0x0 : div1 0x1 : div2 0x2 : div3 . . 0xF = div16 Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register.
15-4	RESERVED	R/W	X	
3-0	GATE	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.25 APP_LIN_CLKSTAT Register (Offset = 60h) [Reset = X]

APP_LIN_CLKSTAT is shown in Table 5-172.

Return to the Summary Table.

Table 5-172. APP_LIN_CLKSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
3-0	CURRDIVR	R	0h	Gives the current divr setting used by the clock divider.

5.2.2.26 RESERVED0 Register (Offset = 64h) [Reset = X]

RESERVED0 is shown in Table 5-173.

Return to the Summary Table.

Table 5-173. RESERVED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WPHRES	R/W	0h	Reserved
23-16	RESERVED	R/W	X	
15-8	RORES	R	0h	Reserved
7-0	RWRES	R/W	0h	Bit 2:0 - Software reset for hwass cba3to2 and m64top32 bridge instances in appss. (reset value - 0) Bit 5:3 - Software reset for fecss p2p async bridge instance in appss. (reset value - 0) 0x0 : Release the reset 0x7 : Assert the reset Bit 31:6 - Reserved

5.2.2.27 RESERVED1 Register (Offset = 68h) [Reset = X]

RESERVED1 is shown in Table 5-174.

Return to the Summary Table.

Table 5-174. RESERVED1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WPHRES	R/W	0h	Reserved
23-16	RESERVED	R/W	X	
15-8	RORES	R	0h	Reserved

Table 5-174. RESERVED1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	RWRES	R/W	0h	Reserved

5.2.2.28 RESERVED2 Register (Offset = 6Ch) [Reset = X]

RESERVED2 is shown in Table 5-175.

Return to the Summary Table.

Table 5-175. RESERVED2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WPHRES	R/W	0h	Reserved
23-16	RESERVED	R/W	X	
15-8	RORES	R	0h	Reserved
7-0	RWRES	R/W	0h	Reserved

5.2.2.29 RESERVED3 Register (Offset = 70h) [Reset = X]

RESERVED3 is shown in Table 5-176.

Return to the Summary Table.

Table 5-176. RESERVED3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	WPHRES	R/W	0h	Reserved
23-16	RESERVED	R/W	X	
15-8	RORES	R	0h	Reserved
7-0	RWRES	R/W	0h	Reserved

5.2.2.30 IPCFGCLKGATE0 Register (Offset = 74h) [Reset = X]

IPCFGCLKGATE0 is shown in Table 5-177.

Return to the Summary Table.

Table 5-177. IPCFGCLKGATE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-27	APP_I2C	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
26-24	APP_DCC	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
23-21	APP_WD	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
20-18	APP_RTI	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
17-15	APP_ESM	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
14-12	TPCC_A	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
11-9	TPTC_A1	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
8-6	TPTC_A0	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

Table 5-177. IPCFGCLKGATE0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-3	APP_QSPI	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
2-0	XBARA	R/W	7h	Reserved Setting this bit does not cause any affect to any logic

5.2.2.31 IPCFGCLKGATE1 Register (Offset = 78h) [Reset = X]

IPCFGCLKGATE1 is shown in Table 5-178.

Return to the Summary Table.

Table 5-178. IPCFGCLKGATE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-27	RES	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
26-24	APP_CTRL	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
23-21	APP_CRC	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
20-18	APP_PWM	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
17-15	APP_LIN	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
14-12	APP_CAN	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
11-9	APP_SPI_1	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
8-6	APP_SPI_0	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
5-3	APP_UART_1	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock
2-0	APP_UART_0	R/W	7h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.32 IPCFGCLKGATE2 Register (Offset = 7Ch) [Reset = X]

IPCFGCLKGATE2 is shown in Table 5-179.

Return to the Summary Table.

Table 5-179. IPCFGCLKGATE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	X	
14-12	PCR6	R/W	0h	0x0 : Enable the Clock 0x7 : Gate the clock
11-9	PCR5	R/W	0h	0x0 : Enable the Clock 0x7 : Gate the clock
8-6	HWASS	R/W	0h	0x0 : Enable the Clock 0x7 : Gate the clock
5-3	RS232	R/W	0h	0x0 : Enable the Clock 0x7 : Gate the clock
2-0	GIO	R/W	0h	0x0 : Enable the Clock 0x7 : Gate the clock

5.2.2.33 BLOCKRESET0 Register (Offset = 80h) [Reset = X]

BLOCKRESET0 is shown in Table 5-180.

Return to the Summary Table.

Table 5-180. BLOCKRESET0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-27	APP_I2C	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
26-24	APP_DCC	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
23-21	APP_WD	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
20-18	APP_RTI	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
17-15	APP_ESM	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
14-12	TPCC_A	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
11-9	TPTC_A1	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
8-6	TPTC_A0	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
5-3	APP_QSPI	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
2-0	HWASS	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset

5.2.2.34 BLOCKRESET1 Register (Offset = 84h) [Reset = X]

BLOCKRESET1 is shown in Table 5-181.

Return to the Summary Table.

Table 5-181. BLOCKRESET1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-27	TOPSS	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
26-24	APP_CTRL	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
23-21	APP_CRC	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
20-18	APP_PWM	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
17-15	APP_LIN	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
14-12	APP_CAN	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
11-9	APP_SPI_1	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
8-6	APP_SPI_0	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
5-3	APP_UART_1	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset

Table 5-181. BLOCKRESET1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	APP_UART_0	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset

5.2.2.35 BLOCKRESET2 Register (Offset = 88h) [Reset = X]

BLOCKRESET2 is shown in Table 5-182.

Return to the Summary Table.

Table 5-182. BLOCKRESET2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8-6	FRC	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
5-3	RS232	R/W	0h	0x0 : Release the reset 0x7 : Assert the reset
2-0	RESERVED	R/W	0h	Reserved

5.2.2.36 PLATFORM_SIGNATURE Register (Offset = 8Ch) [Reset = 7432150Ch]

PLATFORM_SIGNATURE is shown in Table 5-183.

Return to the Summary Table.

Table 5-183. PLATFORM_SIGNATURE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGNATURE	R/W	7432150Ch	Platform signature to identify the platform

5.2.2.37 POWERMODE Register (Offset = 90h) [Reset = X]

POWERMODE is shown in Table 5-184.

Return to the Summary Table.

Table 5-184. POWERMODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	CM3_DEEPSLEEP_STATUS	R	0h	CM3 Core Deep Sleep Status
2	CM3_SLEEP_STATUS	R	0h	CM3 Core Sleep Status
1	DEEPSLEEP	R/W	0h	0x0 : CM4 CORE DEEP SLEEP 0x1 : DEVICE DEEP SLEEP
0	SLEEP	R/W	0h	0x0 : CM4 CORE SLEEP 0x1 : DEVICE SLEEP

5.2.2.38 RST_WFICHECK Register (Offset = 94h) [Reset = X]

RST_WFICHECK is shown in Table 5-185.

Return to the Summary Table.

Table 5-185. RST_WFICHECK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	

Table 5-185. RST_WFICHECK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	CPU	R/W	0h	Writing '000' will disable check for WFI before local reset assertion of app cpu

5.2.2.39 RST_ASSERTDLY Register (Offset = 98h) [Reset = X]

RST_ASSERTDLY is shown in Table 5-186.

Return to the Summary Table.

Table 5-186. RST_ASSERTDLY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	COMMON	R/W	0h	Value decides number of cycles reset should be asserted for cpu

5.2.2.40 RST2ASSERTDLY Register (Offset = 9Ch) [Reset = X]

RST2ASSERTDLY is shown in Table 5-187.

Return to the Summary Table.

Table 5-187. RST2ASSERTDLY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	X	
15-8	CPU	R/W	0h	Value decides number of cycles to be held before asserting reset for app cpu
7-0	RESERVED	R/W	X	

5.2.2.41 RST_FSM_TRIG Register (Offset = A0h) [Reset = X]

RST_FSM_TRIG is shown in Table 5-188.

Return to the Summary Table.

Table 5-188. RST_FSM_TRIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	CPU	R/W	0h	FSM Reset Trigger

5.2.2.42 RST_CAUSE Register (Offset = A4h) [Reset = X]

RST_CAUSE is shown in Table 5-189.

Return to the Summary Table.

Table 5-189. RST_CAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	X	

Table 5-189. RST_CAUSE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	COMMON	R	3h	Reset cause register for APP CPU 0x00 - All cleared 0x01 - Power On Reset (PoR) 0x02 - Subsystem Reset (Combination of Warm Reset initiated from PRCM using xWRLx432:TOP_PRCM:RST_APP_PD_SOFT_RESET and PoR reset) 0x04 - STC RESET 0x08 - Reserved 0x10 - CPU Only Reset triggered by writing to xWRLx432:APP_RCM:RST_FSM_TRIG<RST_FSM_TRIG_CPU> 0x20 - Core Reset initiated from PRCM using xWRLx432:TOP_PRCM:RST_SOFT_APP_CORE_SYSRESET_REQ (reset CPU unconditionally - by debugger) or xWRLx432:TOP_PRCM:APP_CORE_SYSRESET_PARAM_WAKEUP_OUT_STATE 0x40 - Reserved

5.2.2.43 RST_CAUSE_CLR Register (Offset = A8h) [Reset = X]

RST_CAUSE_CLR is shown in Table 5-190.

Return to the Summary Table.

Table 5-190. RST_CAUSE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	CPU	R/W	0h	Writing '111' will clear the RST_CAUSE register

5.2.2.44 XTALCLK_CLK_GATE Register (Offset = ACh) [Reset = X]

XTALCLK_CLK_GATE is shown in Table 5-191.

Return to the Summary Table.

Table 5-191. XTALCLK_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	XTALCLK_CLK_GATE	R/W	0h	RESERVED

5.2.2.45 XTALCLKX2_CLK_GATE Register (Offset = B0h) [Reset = X]

XTALCLKX2_CLK_GATE is shown in Table 5-192.

Return to the Summary Table.

Table 5-192. XTALCLKX2_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	XTALCLKX2_CLK_GATE	R/W	0h	Writing 3'b111 will gate the XTALX2 clock.Writing 3'b000 will ungate the clock.

5.2.2.46 APLLDIV2_CLK_GATE Register (Offset = B4h) [Reset = X]

APLLDIV2_CLK_GATE is shown in Table 5-193.

Return to the Summary Table.

Table 5-193. APLLDIV2_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	APLLDIV2_CLK_GATE	R/W	0h	Writing 3'b111 will gate the APLL/2clock to EDCC.Writing 3'b000 will ungate the clock.

5.2.2.47 DFT_APPSS_LSTC_CLK_GATE Register (Offset = B8h) [Reset = X]

DFT_APPSS_LSTC_CLK_GATE is shown in Table 5-194.

Return to the Summary Table.

Table 5-194. DFT_APPSS_LSTC_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	DFT_APPSS_LSTC_CLK_GATE	R/W	7h	Writing 3'b111 will gate the clock to LSTC.Writing 3'b000 will ungate the clock.

5.2.2.48 DFT_APPSS_LSTC_VBUSP_CLK_GATE Register (Offset = BCh) [Reset = X]

DFT_APPSS_LSTC_VBUSP_CLK_GATE is shown in Table 5-195.

Return to the Summary Table.

Table 5-195. DFT_APPSS_LSTC_VBUSP_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	7h	3'b 000 : Gate clock to LSTC VBUSP 3'b 111 : Ungate Clock to LSTC VBUSP

5.2.2.49 APP_ROM_CLOCK_GATE Register (Offset = C0h) [Reset = X]

APP_ROM_CLOCK_GATE is shown in Table 5-196.

Return to the Summary Table.

Table 5-196. APP_ROM_CLOCK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	3'b 000 : Ungate clock to APP ROM 3'b 111 : Gate Clock to APP ROM

5.2.2.50 APP_RAM1_CLOCK_GATE Register (Offset = C4h) [Reset = X]

APP_RAM1_CLOCK_GATE is shown in Table 5-197.

Return to the Summary Table.

Table 5-197. APP_RAM1_CLOCK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	3'b 000 : Ungate clock to APP RAM1 3'b 111 : Gate Clock to APP_RAM1

5.2.2.51 APP_RAM2_CLOCK_GATE Register (Offset = C8h) [Reset = X]

APP_RAM2_CLOCK_GATE is shown in Table 5-198.

Return to the Summary Table.

Table 5-198. APP_RAM2_CLOCK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	3'b 000 : Ungate clock to APP RAM2 3'b 111 : Gate Clock to APP RAM2

5.2.2.52 APP_RAM3_CLOCK_GATE Register (Offset = CCh) [Reset = X]

APP_RAM3_CLOCK_GATE is shown in Table 5-199.

Return to the Summary Table.

Table 5-199. APP_RAM3_CLOCK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	3'b 000 : Ungate clock to APP RAM3 3'b 111 : Gate Clock to APP RAM3

5.2.2.53 CFG_XBARA_DYNAMIC_CG Register (Offset = D0h) [Reset = X]

CFG_XBARA_DYNAMIC_CG is shown in Table 5-200.

Return to the Summary Table.

Table 5-200. CFG_XBARA_DYNAMIC_CG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	Enable APPSS crossbar dynamic clock gating. 1 - Dynamic clock gating feature is enabled. When the feature is enabed, the CM4 should monitor for any possible pending transactions from various masters like DMA/NWA and if no transaction is expected to be iniated by the masters, the CM4 executes WFI. On ssertion of WFI signal, the clock to crossbar is gated. The clock is automatically ungated under two conditions (i) when the WFI signal is deasserted by any interrupted (ii) when any of the APPSS TPCC triggers are asserted. Instead of WFI, cfg_xbara_set_dynamic_cg also can be used to start the clock gating. 0 - Dynamic clock gating feature is disabled. The clock to APPSS crossbar is not gated dynamically. The clock to APPSS crossbar is gated/ungated as per device ice level power states.

5.2.2.54 CFG_TPTC1_DYNAMIC_CG Register (Offset = D4h) [Reset = X]

CFG_TPTC1_DYNAMIC_CG is shown in Table 5-201.

Return to the Summary Table.

Table 5-201. CFG_TPTC1_DYNAMIC_CG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	

Table 5-201. CFG_TPTC1_DYNAMIC_CG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	ENABLE	R/W	0h	Enable APPSS TPTC1 crossbar dynamic clock gating. 1 - Dynamic clock gating feature is enabled. Same behaviour as cfg_xbara_dynamic_cg_en - for both entry to clock gating and exit from clock gating. WFI or cfg_tptc1_set_dynamic_cg 0 - Dynamic clock gating feature is disabled. The clock to APPSS TPTC1 crossbar is not gated dynamically. The clock to APPSS TPTC1 crossbar is gated/ungated as per device ice level power states.

5.2.2.55 CFG_TPTC2_DYNAMIC_CG Register (Offset = D8h) [Reset = X]

CFG_TPTC2_DYNAMIC_CG is shown in Table 5-202.

Return to the Summary Table.

Table 5-202. CFG_TPTC2_DYNAMIC_CG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	Enable APPSS TPTC2 crossbar dynamic clock gating. 1 - Dynamic clock gating feature is enabled. Same behaviour as cfg_xbara_dynamic_cg_en - for both entry to clock gating and exit from clock gating. WFI or cfg_tptc2_set_dynamic_cg 0 - Dynamic clock gating feature is disabled. The clock to APPSS TPTC2 crossbar is not gated dynamically. The clock to APPSS TPTC2 crossbar is gated/ungated as per device ice level power states

5.2.2.56 CFG_XBARA_SET_DYNAMIC_CG Register (Offset = DCh) [Reset = X]

CFG_XBARA_SET_DYNAMIC_CG is shown in Table 5-203.

Return to the Summary Table.

Table 5-203. CFG_XBARA_SET_DYNAMIC_CG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	SET	R/W	0h	Start APPSS crossbar dynamic clock gating. This is used instead of WFI. 1 - Start the clock gating. In order to start again, write 0 followed by 1. Rise edge is detected internally, to start the clock gating. 0 - Clock is ungated. Fall edge is detected internally to ungate the clock.

5.2.2.57 CFG_TPTC1_SET_DYNAMIC_CG Register (Offset = E0h) [Reset = X]

CFG_TPTC1_SET_DYNAMIC_CG is shown in Table 5-204.

Return to the Summary Table.

Table 5-204. CFG_TPTC1_SET_DYNAMIC_CG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	

Table 5-204. CFG_TPTC1_SET_DYNAMIC_CG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SET	R/W	0h	Start APPSS tptc1 dynamic clock gating. This is used instead of WFI. 1 - Start the clock gating. In order to start again, write 0 followed by 1. Rise edge is detected internally, to start the clock gating. 0 - Clock is ungated. Fall edge is detected internally to ungate the clock.

5.2.2.58 CFG_TPTC2_SET_DYNAMIC_CG Register (Offset = E4h) [Reset = X]

CFG_TPTC2_SET_DYNAMIC_CG is shown in Table 5-205.

Return to the Summary Table.

Table 5-205. CFG_TPTC2_SET_DYNAMIC_CG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	SET	R/W	0h	Start APPSS tptc2 dynamic clock gating. This is used instead of WFI. 1 - Start the clock gating. In order to start again, write 0 followed by 1. Rise edge is detected internally, to start the clock gating. 0 - Clock is ungated. Fall edge is detected internally to ungate the clock.

5.2.2.59 CM4_FORCE_HCLK_GATE Register (Offset = E8h) [Reset = X]

CM4_FORCE_HCLK_GATE is shown in Table 5-206.

Return to the Summary Table.

Table 5-206. CM4_FORCE_HCLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	3'b 000 - CM4 HCLK is ungated 3'b 111 - CM4 HCLK is gated

5.2.2.60 LIN_SCI_DIV Register (Offset = ECh) [Reset = X]

LIN_SCI_DIV is shown in Table 5-207.

Return to the Summary Table.

Table 5-207. LIN_SCI_DIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	VAL	R/W	0h	ICG Based Divider for LIN

5.2.2.61 APP_LSTC_EN Register (Offset = F0h) [Reset = X]

APP_LSTC_EN is shown in Table 5-208.

Return to the Summary Table.

Table 5-208. APP_LSTC_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ENABLE	R/W	1h	Enable vbusp_req and clk_en for app lstc

5.2.2.62 LOCK0_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0_KICK0 is shown in Table 5-209.

Return to the Summary Table.

- KICK0 component

Table 5-209. LOCK0_KICK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_KICK0	R/W	0h	- KICK0 component

5.2.2.63 LOCK0_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0_KICK1 is shown in Table 5-210.

Return to the Summary Table.

- KICK1 component

Table 5-210. LOCK0_KICK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_KICK1	R/W	0h	- KICK1 component

5.2.2.64 INTR_RAW_STATUS Register (Offset = 1010h) [Reset = X]

INTR_RAW_STATUS is shown in Table 5-211.

Return to the Summary Table.

Interrupt Raw Status/Set Register

Table 5-211. INTR_RAW_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	PROXY_ERR	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	KICK_ERR	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	ADDR_ERR	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	PROT_ERR	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

5.2.2.65 INTR_ENABLED_STATUS_CLEAR Register (Offset = 1014h) [Reset = X]

INTR_ENABLED_STATUS_CLEAR is shown in Table 5-212.

Return to the Summary Table.

Interrupt Enabled Status/Clear register

Table 5-212. INTR_ENABLED_STATUS_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	ENABLED_PROXY_ERR	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	ENABLED_KICK_ERR	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	ENABLED_ADDR_ERR	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	ENABLED_PROT_ERR	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

5.2.2.66 INTR_ENABLE Register (Offset = 1018h) [Reset = X]

INTR_ENABLE is shown in Table 5-213.

Return to the Summary Table.

Interrupt Enable register

Table 5-213. INTR_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	PROXY_ERR_EN	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	KICK_ERR_EN	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	ADDR_ERR_EN	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	PROT_ERR_EN	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

5.2.2.67 INTR_ENABLE_CLEAR Register (Offset = 101Ch) [Reset = X]

INTR_ENABLE_CLEAR is shown in Table 5-214.

Return to the Summary Table.

Interrupt Enable Clear register

Table 5-214. INTR_ENABLE_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	PROXY_ERR_EN_CLR	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	KICK_ERR_EN_CLR	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	ADDR_ERR_EN_CLR	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	PROT_ERR_EN_CLR	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

5.2.2.68 EOI Register (Offset = 1020h) [Reset = X]

EOI is shown in Table 5-215.

Return to the Summary Table.

EOI register

Table 5-215. EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	EOI_VECTOR	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

5.2.2.69 FAULT_ADDRESS Register (Offset = 1024h) [Reset = 00000000h]

FAULT_ADDRESS is shown in Table 5-216.

Return to the Summary Table.

Fault Address register

Table 5-216. FAULT_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FAULT_ADDR	R	0h	Fault Address.

5.2.2.70 FAULT_TYPE_STATUS Register (Offset = 1028h) [Reset = X]

FAULT_TYPE_STATUS is shown in Table 5-217.

Return to the Summary Table.

Fault Type Status register

Table 5-217. FAULT_TYPE_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	X	
6	FAULT_NS	R	0h	Non-secure access.

Table 5-217. FAULT_TYPE_STATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	FAULT_TYPE	R	0h	Fault Type 10_ 0000 = Supervisor read fault - priv = 1 dir = 1 dtype != 1 01_ 0000 = Supervisor write fault - priv = 1 dir = 0 00_ 1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_ 0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_ 0010 = User write fault - priv = 0 dir = 0 00_ 0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_ 0000 = No fault

5.2.2.71 FAULT_ATTR_STATUS Register (Offset = 102Ch) [Reset = 00000000h]

FAULT_ATTR_STATUS is shown in Table 5-218.

Return to the Summary Table.

Fault Attribute Status register

Table 5-218. FAULT_ATTR_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	FAULT_XID	R	0h	XID.
19-8	FAULT_ROUTEID	R	0h	Route ID.
7-0	FAULT_PRIVID	R	0h	Privilege ID.

5.2.2.72 FAULT_CLEAR Register (Offset = 1030h) [Reset = X]

FAULT_CLEAR is shown in Table 5-219.

Return to the Summary Table.

Fault Clear register

Table 5-219. FAULT_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	X	
0	FAULT_CLR	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

5.2.3 APP_CTRL Registers

Table 5-220 lists the memory-mapped registers for the APP_CTRL registers. All register offset addresses not listed in Table 5-220 should be considered as reserved locations and the register contents should not be modified.

Table 5-220. APP_CTRL Registers

Offset	Acronym	Register Name	Section
0h	PID	PID register	Go
4h	HW_REG0		Go
8h	HW_REG1		Go
Ch	PREVIOUS_NAME		Go
10h	HW_REG3		Go
14h	HW_REG4		Go
18h	HW_REG5		Go
1Ch	HW_REG6		Go
20h	HW_REG7		Go
24h	APPSS_SW_INT		Go
28h	APPSS_IPC_RFS		Go
2Ch	APPSS_CAPEVNT_SEL		Go
30h	APPSS_DMA_REQ_SEL		Go
34h	APPSS_DMA1_REQ_SEL		Go
38h	APPSS_IRQ_REQ_SEL		Go
3Ch	APPSS_SPI_TRIG_SRC		Go
40h	APPSS_RAM1A_MEM_INIT		Go
44h	APPSS_RAM1A_MEM_INIT_DONE		Go
48h	APPSS_RAM1A_MEM_INIT_STATUS		Go
4Ch	APPSS_RAM2A_MEM_INIT		Go
50h	APPSS_RAM2A_MEM_INIT_DONE		Go
54h	APPSS_RAM2A_MEM_INIT_STATUS		Go
58h	APPSS_RAM3A_MEM_INIT		Go
5Ch	APPSS_RAM3A_MEM_INIT_DONE		Go
60h	APPSS_RAM3A_MEM_INIT_STATUS		Go
64h	HWASS_SHRD_RAM0_MEM_INIT		Go
68h	HWASS_SHRD_RAM0_MEM_INIT_DONE		Go
6Ch	HWASS_SHRD_RAM0_MEM_INIT_STAT US		Go
70h	HWASS_SHRD_RAM1_MEM_INIT		Go
74h	HWASS_SHRD_RAM1_MEM_INIT_DONE		Go
78h	HWASS_SHRD_RAM1_MEM_INIT_STAT US		Go
7Ch	APPSS_TPCC_MEMINIT_START		Go
80h	APPSS_TPCC_MEMINIT_DONE		Go
84h	APPSS_TPCC_MEMINIT_STATUS		Go
88h	APPSS_SPIA_CFG		Go
8Ch	APPSS_SPIB_CFG		Go
90h	APPSS_EPWM_CFG		Go
94h	RESERVED		Go
98h	APPSS_MCAN_FE_AND_LIN_INTR_SEL		Go
9Ch	APPSS_MCANA_INT_CLR		Go
A0h	APPSS_MCANA_INT_MASK		Go

Table 5-220. APP_CTRL Registers (continued)

Offset	Acronym	Register Name	Section
A4h	APPSS_MCANA_INT_STAT		Go
A8h	APPSS_CM4_GLOBAL_CONFIG		Go
ACh	RESERVED1		Go
B8h	APPSS_AHB_CTRL		Go
BCh	ESM_GATING0		Go
C0h	ESM_GATING1		Go
C4h	ESM_GATING2		Go
C8h	ESM_GATING3		Go
CCh	ESM_GATING4		Go
D0h	ESM_GATING5		Go
D4h	ESM_GATING6		Go
D8h	ESM_GATING7		Go
DCh	APPSS_CM4_HALT		Go
E0h	APPSS_CM4_EVENT		Go
E4h	SPIA_IO_CFG		Go
E8h	SPIB_IO_CFG		Go
ECh	SPI_HOST_IRQ		Go
F0h	TPTC_DBS_CONFIG		Go
F4h	TPCC_PARITY_CTRL		Go
F8h	TPCC_PARITY_STATUS		Go
FCh	APPSS_DBG_ACK_CTL0		Go
100h	DEBUGSS_CSETB_FLUSH		Go
104h	CPSW_CONTROL		Go
108h	APPSS_ERRAGG_MASK0		Go
10Ch	APPSS_ERRAGG_STATUS0		Go
190h	APPSS_TPCC_A_ERRAGG_MASK		Go
194h	APPSS_TPCC_A_ERRAGG_STATUS		Go
198h	APPSS_TPCC_A_ERRAGG_STATUS_RAW		Go
214h	APPSS_TPCC_A_INTAGG_MASK		Go
218h	HW_SPARE_WPH		Go
21Ch	APPSS_TPCC_A_INTAGG_STATUS_RAW		Go
274h	APPSS_TPCC_B_ERRAGG_MASK		Go
278h	APPSS_TPCC_B_ERRAGG_STATUS		Go
27Ch	APPSS_TPCC_B_ERRAGG_STATUS_RAW		Go
2ECh	APPSS_TPCC_B_INTAGG_MASK		Go
2F0h	APPSS_TPCC_B_INTAGG_STATUS		Go
2F4h	APPSS_TPCC_B_INTAGG_STATUS_RAW		Go
2F8h	APPSS_MPU_ERRAGG_MASK		Go
2FCh	APPSS_MPU_ERRAGG_STATUS		Go
300h	APPSS_MPU_ERRAGG_STATUS_RAW		Go
304h	APPSS_QSPI_CONFIG		Go
308h	APPSS_CTI_TRIG_SEL		Go
30Ch	APPSS_DBGSS_CTI_TRIG_SEL		Go
310h	APPSS_BOOT_INFO_REG0		Go

Table 5-220. APP_CTRL Registers (continued)

Offset	Acronym	Register Name	Section
314h	APPSS_BOOT_INFO_REG1		Go
318h	APPSS_BOOT_INFO_REG2		Go
31Ch	APPSS_BOOT_INFO_REG3		Go
320h	APPSS_BOOT_INFO_REG4		Go
324h	APPSS_BOOT_INFO_REG5		Go
328h	APPSS_BOOT_INFO_REG6		Go
32Ch	APPSS_BOOT_INFO_REG7		Go
330h	APPSS_TPTC_ECCAGGR_CLK_CNTRL		Go
334h	APPSS_TPTC_BOUNDARY_CFG		Go
338h	APPSS_TPTC_XID_REORDER_CFG		Go
33Ch	HW_SYNC_FE_CTRL		Go
340h	HW_SPARE_REG1		Go
344h	HW_SPARE_REG2		Go
348h	HW_SPARE_REG3		Go
34Ch	NERROR_MASK		Go
350h	HW_SPARE_RW0		Go
354h	HW_SPARE_RW1		Go
358h	HW_SPARE_RW2		Go
35Ch	HW_SPARE_RW3		Go
360h	HW_SPARE_RW4		Go
364h	HW_SPARE_RW5		Go
368h	HW_SPARE_RO0		Go
36Ch	HW_SPARE_RO1		Go
370h	HW_SPARE_RO2		Go
374h	HW_SPARE_RO3		Go
378h	HW_SPARE_REC		Go
37Ch	APP_CTRL		Go
380h	WIC_CTRL		Go
384h	WIC_STAT_CLR		Go
388h	WIC_STAT		Go
38Ch	WICEN		Go
390h	FORCEFCLKACTIVE		Go
394h	FECSS_CLK_GATE		Go
398h	APPSS_SHARED_MEM_CLK_GATE		Go
39Ch	APPSS_MEM_INIT_SLICE_SEL		Go
3A0h	APPSS_QSPI_CHAR_EXT_CLK_EN		Go
3A4h	APPSS_QSPI_EXT_CLK_EN		Go
3A8h	SPI1_SMART_IDLE		Go
3ACh	SPI2_SMART_IDLE		Go
3B0h	CAN_SMART_IDLE		Go
3B4h	LIN_SMART_IDLE		Go
3B8h	HWASS_CLK_GATE		Go
3BCh	CFG_TIMEOUT_PCR3		Go
3C0h	RESERVED0		Go
3C4h	APPSS_ERRAGG_MASK1		Go
3C8h	APPSS_ERRAGG_STATUS1		Go
3CCh	FORCEHCLKACTIVE		Go

Table 5-220. APP_CTRL Registers (continued)

Offset	Acronym	Register Name	Section
3D0h	APPSS_RAM1_OWRITE_ERR		Go
3D4h	APPSS_RAM1_OWRITE_ERR_ADDR		Go
3D8h	APPSS_RAM2_OWRITE_ERR		Go
3DCh	APPSS_RAM2_OWRITE_ERR_ADDR		Go
3E0h	APPSS_RAM3_OWRITE_ERR		Go
3E4h	APPSS_RAM3_OWRITE_ERR_ADDR		Go
3E8h	APPSS_SHRD_RAM_OWRITE_ERR		Go
3ECh	APPSS_SHRD_RAM_OWRITE_ERR_ADDR		Go
3F0h	APPSS_OWRITE_ERR_AGGR		Go
3F4h	HW_SPARE_RW6		Go
3F8h	HW_SPARE_RW7		Go
3FCh	HW_SPARE_RW8		Go
400h	HW_SPARE_RW9		Go
404h	HW_SPARE_HWA_RW0		Go
1008h	LOCK0_KICK0	- KICK0 component	Go
100Ch	LOCK0_KICK1	- KICK1 component	Go
1010h	INTR_RAW_STATUS	Interrupt Raw Status/Set Register	Go
1014h	INTR_ENABLED_STATUS_CLEAR	Interrupt Enabled Status/Clear register	Go
1018h	INTR_ENABLE	Interrupt Enable register	Go
101Ch	INTR_ENABLE_CLEAR	Interrupt Enable Clear register	Go
1020h	EOI	EOI register	Go
1024h	FAULT_ADDRESS	Fault Address register	Go
1028h	FAULT_TYPE_STATUS	Fault Type Status register	Go
102Ch	FAULT_ATTR_STATUS	Fault Attribute Status register	Go
1030h	FAULT_CLEAR	Fault Clear register	Go

Complex bit access types are encoded to fit into small table cells. Table 5-221 shows the codes that are used for access types in this section.

Table 5-221. APP_CTRL Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

5.2.3.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in Table 5-222.

Return to the Summary Table.

PID register

Table 5-222. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PID_MSB16	R	6180h	
15-11	PID_MISC	R	0h	
10-8	PID_MAJOR	R	2h	
7-6	PID_CUSTOM	R	0h	
5-0	PID_MINOR	R	14h	

5.2.3.2 HW_REG0 Register (Offset = 4h) [Reset = 00000000h]

HW_REG0 is shown in Table 5-223.

Return to the Summary Table.

Table 5-223. HW_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG0	R/W	0h	HW reserved Regsiter

5.2.3.3 HW_REG1 Register (Offset = 8h) [Reset = 00000000h]

HW_REG1 is shown in Table 5-224.

Return to the Summary Table.

Table 5-224. HW_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG1	R/W	0h	HW reserved Regsiter

5.2.3.4 PREVIOUS_NAME Register (Offset = Ch) [Reset = 00000000h]

PREVIOUS_NAME is shown in Table 5-225.

Return to the Summary Table.

Table 5-225. PREVIOUS_NAME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG2	R/W	0h	HW reserved Regsiter

5.2.3.5 HW_REG3 Register (Offset = 10h) [Reset = 00000000h]

HW_REG3 is shown in Table 5-226.

Return to the Summary Table.

Table 5-226. HW_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG3	R/W	0h	HW reserved Regsiter

5.2.3.6 HW_REG4 Register (Offset = 14h) [Reset = 00000000h]

HW_REG4 is shown in Table 5-227.

Return to the Summary Table.

Table 5-227. HW_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG4	R/W	0h	HW reserved Regsiter

5.2.3.7 HW_REG5 Register (Offset = 18h) [Reset = 00000000h]

HW_REG5 is shown in Table 5-228.

Return to the Summary Table.

Table 5-228. HW_REG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG5	R/W	0h	HW reserved Regsiter

5.2.3.8 HW_REG6 Register (Offset = 1Ch) [Reset = 00000000h]

HW_REG6 is shown in Table 5-229.

Return to the Summary Table.

Table 5-229. HW_REG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG6	R/W	0h	HW reserved Regsiter

5.2.3.9 HW_REG7 Register (Offset = 20h) [Reset = 00000000h]

HW_REG7 is shown in Table 5-230.

Return to the Summary Table.

Table 5-230. HW_REG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWREG7	R/W	0h	HW reserved Regsiter

5.2.3.10 APPSS_SW_INT Register (Offset = 24h) [Reset = X]

APPSS_SW_INT is shown in Table 5-231.

Return to the Summary Table.

Table 5-231. APPSS_SW_INT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-0	PULSE	R/W	0h	Write_pulse bit field: writing 1'b1 to each bit will trigger SW_INT< 0-3> respectively to CM4.

5.2.3.11 APPSS_IPC_RFS Register (Offset = 28h) [Reset = 00000000h]

APPSS_IPC_RFS is shown in Table 5-232.

Return to the Summary Table.

Table 5-232. APPSS_IPC_RFS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	COMMAND	R/W	0h	Used by software to communicate commands and response. It is 7-bits per interrupt.

Table 5-232. APPSS_IPC_RFS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	HOST_INTR	R/W	0h	Write_pulse bit field: Writing 1'b1 to each bit will trigger HOST_INTR 0-3> respectively to CM3.

5.2.3.12 APPSS_CAPEVNT_SEL Register (Offset = 2Ch) [Reset = X]

APPSS_CAPEVNT_SEL is shown in Table 5-233.

Return to the Summary Table.

Table 5-233. APPSS_CAPEVNT_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	
23-12	SRC1	R/W	0h	5: 0 : Selects the interrupt to route to RTI Capture event 1 6'd 0 : MUXED_FECSS_CHIRPTIMER_CHIRP_START_AND_CHIRP_END 6'd 1 : MUXED_FECSS_CHIRPTIMER_BURST_START_AND_BURST_END 6'd 2 : FECSS_CHIRPTIMER_FRAME_END 6'd 3 : FECSS_FRAMETIMER_FRAME_START 6'd 4 : MUXED_FECSS_CHIRP_AVAIL_IRQ_AND_ADC_VALID_START_AND_SYNC_IN 6'd 5 : MUXED_FECSS_FRAME_START_OFFSET_INTR_TIME1 6'd 6 : FECSS_FRAME_START_OFFSET_INTR_TIME2 6'd 7 : FECSS_FRAME_START_OFFSET_INTR_TIME3 6'd 8 : FECSS_BURST_START_OFFSET_TIME 11: 6: Selects the interrupt to route to WDT Capture event 1 6'd 0 : MUXED_FECSS_CHIRPTIMER_CHIRP_START_AND_CHIRP_END 6'd 1 : MUXED_FECSS_CHIRPTIMER_BURST_START_AND_BURST_END 6'd 2 : FECSS_CHIRPTIMER_FRAME_END 6'd 3 : FECSS_FRAMETIMER_FRAME_START 6'd 4 : MUXED_FECSS_CHIRP_AVAIL_IRQ_AND_ADC_VALID_START_AND_SYNC_IN 6'd 5 : MUXED_FECSS_FRAME_START_OFFSET_INTR_TIME1 6'd 6 : FECSS_FRAME_START_OFFSET_INTR_TIME2 6'd 7 : FECSS_FRAME_START_OFFSET_INTR_TIME3 6'd 8 : FECSS_BURST_START_OFFSET_TIME 6'd 9 : FECSS_IPC_RFS_FEC_INTR 1

Table 5-233. APPSS_CAPEVNT_SEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-0	SRC0	R/W	0h	5: 0 : Selects the interrupt to route to RTI Capture event 0 6'd 0 : FECSS_FRAME_START_OFFSET_INTR_TIME3 6'd 1 : FECSS_FRAME_START_OFFSET_INTR_TIME2 6'd 2 : FECSS_FRAME_START_OFFSET_INTR_TIME1 6'd 3 : FECSS_FRAMETIMER_FRAME_START 6'd 4 : LIN_INT1 6'd 5 : LIN_INT0 6'd 6 : MCAN_FE_INT7 6'd 7 : MCAN_FE_INT1 6'd 8 : MCAN_FE_INT2 6'd 9 : MCAN_FE_INT3 6'd 10 : MCAN_FE_INT4 6'd 11 : MCAN_FE_INT5 6'd 12 : MCAN_FE_INT6 6'd 13 : MCAN_INT0 6'd 14 : MCAN_INT1 6'd 15 : SYNC_IN 11: 6: Selects the interrupt to route to WDT Capture event 0 6'd 0 : MUXED_FECSS_CHIRPTIMER_CHIRP_START_AND_CHIRP_END 6'd 1 : MUXED_FECSS_CHIRPTIMER_BURST_START_AND_BURST_END 6'd 2 : FECSS_CHIRPTIMER_FRAME_END 6'd 3 : FECSS_FRAMETIMER_FRAME_START 6'd 4 : MUXED_FECSS_CHIRP_AVAIL_IRQ_AND_ADC_VALID_START_AND_SYNC_IN 6'd 5 : MUXED_FECSS_FRAME_START_OFFSET_INTR_TIME1 6'd 6 : FECSS_FRAME_START_OFFSET_INTR_TIME2 6'd 7 : FECSS_FRAME_START_OFFSET_INTR_TIME3 6'd 8 : FECSS_BURST_START_OFFSET_TIME 6'd 9 : FECSS_IPC_RFS_FEC_INTR 0

ADVANCE INFORMATION

5.2.3.13 APPSS_DMA_REQ_SEL Register (Offset = 30h) [Reset = 00000000h]

APPSS_DMA_REQ_SEL is shown in Table 5-234.

Return to the Summary Table.

Table 5-234. APPSS_DMA_REQ_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELECT	R/W	0h	Reserved for R&D. Do not touch

5.2.3.14 APPSS_DMA1_REQ_SEL Register (Offset = 34h) [Reset = 00000000h]

APPSS_DMA1_REQ_SEL is shown in Table 5-235.

Return to the Summary Table.

Table 5-235. APPSS_DMA1_REQ_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELECT	R/W	0h	Reserved for R&D. Do not touch

5.2.3.15 APPSS_IRQ_REQ_SEL Register (Offset = 38h) [Reset = 00000000h]

APPSS_IRQ_REQ_SEL is shown in Table 5-236.

Return to the Summary Table.

Table 5-236. APPSS_IRQ_REQ_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SELECT	R/W	0h	Configuration register APPSS_IRQ_REQ_SEL is used to select the interrupt for CM4. Below are the bit definitions 0 : 0x0 = Map 0th IRQ from compare block of RTI (RTI1) to IRQ43 : 0x1 = Map 0th IRQ from compare block of WDT (RTI2) to IRQ43 1 : 0x0 = Map 1st IRQ from compare block of RTI (RTI1) to IRQ44 : 0x1 = Map 1st IRQ from compare block of WDT (RTI2) to IRQ44 2 : 0x0 = Map 2nd IRQ from compare block of RTI (RTI1) to IRQ45 : 0x1 = Map 2nd IRQ from compare block of WDT (RTI2) to IRQ45 3 : 0x0 = Map 3rd IRQ from compare block of RTI (RTI1) to IRQ46 : 0x1 = Map 3rd IRQ from compare block of WDT (RTI2) to IRQ46 5:4 : 0x00 = Selects time base IRQ from RTI (RTI1) to IRQ47 : 0x01 = Selects time base IRQ from WDT (RTI2) to IRQ47 : 0x10 = Selects gpadc_ifm_done to IRQ47 : 0x11 = Reserved 7:6 : 0x00 = Selects capture event 0 of RTI to IRQ48 and Selects capture event 1 of IRQ from RTI to IRQ49 : 0x01 = Selects capture event 0 of WDT to IRQ48 and Selects capture event 1 of IRQ from WDT to IRQ49 : 0x10 = Selects PWM Interrupt 0 to IRQ48 and PWM Interrupt 1 to IRQ49 : 0x11 = Selects OVL_REQ of RTI (RTI1) to IRQ48 and OVL_REQ of WDT to IRQ49 8 : 0x0 = mcan_fe_int7 connected to IRQ29 : 0x1 = debugss txdata_available interrupt connected to IRQ29. 9 : 0x0 = Used for TPPCA trigger. Dma read interrupt of SPI1/A channel routed to TPCCA (DMA) trigger 62 : 0x1 = Used for TPPCA trigger. dma read interrupt of SPI2/B channel routed to TPCCA (DMA) trigger 62 10 : 0x0 = Used for TPPCA trigger. dma write interrupt of SPI1/A channel routed to TPCCA (DMA) trigger 63 : 0x1 = Used for TPPCA trigger. dma write interrupt of SPI2/B channel routed to TPCCA (DMA) trigger 63 11 : 0x0 = Timing Engine Chirptimer_chirp_start to IRQ30 : 0x1 = Timing Engine Chirptimer_chirp_end to IRQ30 12 : 0x0 = Timing Engine Chirptimer_burst_start to IRQ31 : 0x1 = Timing Engine Chirptimer_burst_end to IRQ31 14:13: 0x00 = chirp_avail_irq to IRQ34 : 0x01 = adc_valid_start to IRQ34 : 0x10 = SYNC_in to IRQ34 15 : Reserved 16 : 0x0 = mcan_fe_int6 connected to IRQ28 : 0x1 = spi2_int_req connected to IRQ28 18:17: 0x00 = DCC_DONE Interrupt connected to IRQ12 : 0x01 = CM4 LBIST Interrupt connected to IRQ12 : 0x10 = CM3 LBIST Interrupt connected to IRQ12 : 0x11 = TOP PBIST Interrupt connected to IRQ12 20:19: 0x00 = I2C_INT Interrupt connected to IRQ19 : 0x01 = CM4 LBIST Interrupt connected to IRQ19 : 0x10 = CM3 LBIST Interrupt connected to IRQ19 : 0x11 = TOP PBIST Interrupt connected to IRQ19 21 : 0x0 = APPSS_TPCC1_ERRAGG connected to IRQ16 : 0x1 = CTI_TRIGOUT2 connected to IRQ16 22 : 0x0 = APPSS_TPCC2_ERRAGG connected to IRQ18 : 0x1 = CTI_TRIGOUT3 connected to IRQ18

5.2.3.16 APPSS_SPI_TRIG_SRC Register (Offset = 3Ch) [Reset = X]

APPSS_SPI_TRIG_SRC is shown in Table 5-237.

Return to the Summary Table.

Table 5-237. APPSS_SPI_TRIG_SRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26-16	TRIG_SPIB	R/W	0h	RESERVED
15-2	RESERVED	R/W	X	
1-0	TRIG_SPIA	R/W	0h	RESERVED

5.2.3.17 APPSS_RAM1A_MEM_INIT Register (Offset = 40h) [Reset = X]

APPSS_RAM1A_MEM_INIT is shown in Table 5-238.

Return to the Summary Table.

Table 5-238. APPSS_RAM1A_MEM_INIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initialization of APPSS_RAM1 partion0 bank. Value in each row is initialized to 0x0C_0000_0000

5.2.3.18 APPSS_RAM1A_MEM_INIT_DONE Register (Offset = 44h) [Reset = X]

APPSS_RAM1A_MEM_INIT_DONE is shown in Table 5-239.

Return to the Summary Table.

Table 5-239. APPSS_RAM1A_MEM_INIT_DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT_DONE	R/W	0h	This field will be high once initialization of APPSS_RAM1 partion0 banks is finished. Writing '1' would clear the bit.

5.2.3.19 APPSS_RAM1A_MEM_INIT_STATUS Register (Offset = 48h) [Reset = X]

APPSS_RAM1A_MEM_INIT_STATUS is shown in Table 5-240.

Return to the Summary Table.

Table 5-240. APPSS_RAM1A_MEM_INIT_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	X	
0	MEM_STATUS	R	0h	1'b 0: No initialization is happening for APPSS_RAM1 partion0 bank 1'b 1: Initialization is in progress for APPSS_RAM1 partion0 bank

5.2.3.20 APPSS_RAM2A_MEM_INIT Register (Offset = 4Ch) [Reset = X]

APPSS_RAM2A_MEM_INIT is shown in Table 5-241.

Return to the Summary Table.

Table 5-241. APPSS_RAM2A_MEM_INIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	

Table 5-241. APPSS_RAM2A_MEM_INIT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	MEM_INIT	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initialization of APPSS_RAM2 partion0 bank. Value in each row is initialized to 0x0C_0000_0000

5.2.3.21 APPSS_RAM2A_MEM_INIT_DONE Register (Offset = 50h) [Reset = X]

APPSS_RAM2A_MEM_INIT_DONE is shown in Table 5-242.

Return to the Summary Table.

Table 5-242. APPSS_RAM2A_MEM_INIT_DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT_DONE	R/W	0h	This field will be high once initialization of APPSS_RAM2 partion0 banks is finished. Writing '1' would clear the bit.

5.2.3.22 APPSS_RAM2A_MEM_INIT_STATUS Register (Offset = 54h) [Reset = X]

APPSS_RAM2A_MEM_INIT_STATUS is shown in Table 5-243.

Return to the Summary Table.

Table 5-243. APPSS_RAM2A_MEM_INIT_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	X	
0	MEM_STATUS	R	0h	1'b 0: No initialization is happening for APPSS_RAM2 partion0 bank 1'b 1: Initialization is in progress for APPSS_RAM2 partion0 bank

5.2.3.23 APPSS_RAM3A_MEM_INIT Register (Offset = 58h) [Reset = X]

APPSS_RAM3A_MEM_INIT is shown in Table 5-244.

Return to the Summary Table.

Table 5-244. APPSS_RAM3A_MEM_INIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initialization of APPSS_RAM3 partion0 bank. Value in each row is initialized to 0x0C_0000_0000

5.2.3.24 APPSS_RAM3A_MEM_INIT_DONE Register (Offset = 5Ch) [Reset = X]

APPSS_RAM3A_MEM_INIT_DONE is shown in Table 5-245.

Return to the Summary Table.

Table 5-245. APPSS_RAM3A_MEM_INIT_DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT_DONE	R/W	0h	This field will be high once initialization of APPSS_RAM3 partion0 banks is finished. Writing '1' would clear the bit.

5.2.3.25 APPSS_RAM3A_MEM_INIT_STATUS Register (Offset = 60h) [Reset = X]

APPSS_RAM3A_MEM_INIT_STATUS is shown in Table 5-246.

Return to the Summary Table.

Table 5-246. APPSS_RAM3A_MEM_INIT_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	X	
0	MEM_STATUS	R	0h	1'b 0: No initialization is happening for APPSS_RAM3 partion0 bank 1'b 1: Initialization is in progress for APPSS_RAM3 partion0 bank

5.2.3.26 HWASS_SHRD_RAM0_MEM_INIT Register (Offset = 64h) [Reset = X]

HWASS_SHRD_RAM0_MEM_INIT is shown in Table 5-247.

Return to the Summary Table.

Table 5-247. HWASS_SHRD_RAM0_MEM_INIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the HWASS Shared RAM parition0. Value in each row is initialized to 0x0

5.2.3.27 HWASS_SHRD_RAM0_MEM_INIT_DONE Register (Offset = 68h) [Reset = X]

HWASS_SHRD_RAM0_MEM_INIT_DONE is shown in Table 5-248.

Return to the Summary Table.

Table 5-248. HWASS_SHRD_RAM0_MEM_INIT_DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT_DONE	R/W	0h	This field will be high once intialization of HWASS Shared RAM parition0 is finished. Writing '1' would clear the bit

5.2.3.28 HWASS_SHRD_RAM0_MEM_INIT_STATUS Register (Offset = 6Ch) [Reset = X]

HWASS_SHRD_RAM0_MEM_INIT_STATUS is shown in Table 5-249.

Return to the Summary Table.

Table 5-249. HWASS_SHRD_RAM0_MEM_INIT_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	X	
0	MEM_STATUS	R	0h	1'b 0: No initialization is happening for HWASS Shared RAM parition0 1'b 1: Initialization is in progress for HWASS Shared RAM parition0

5.2.3.29 HWASS_SHRD_RAM1_MEM_INIT Register (Offset = 70h) [Reset = X]

HWASS_SHRD_RAM1_MEM_INIT is shown in Table 5-250.

Return to the Summary Table.

Table 5-250. HWASS_SHRD_RAM1_MEM_INIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the HWASS Shared RAM parition1. Value in each row is initialized to 0x0

5.2.3.30 HWASS_SHRD_RAM1_MEM_INIT_DONE Register (Offset = 74h) [Reset = X]

HWASS_SHRD_RAM1_MEM_INIT_DONE is shown in Table 5-251.

Return to the Summary Table.

Table 5-251. HWASS_SHRD_RAM1_MEM_INIT_DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MEM_INIT_DONE	R/W	0h	This field will be high once intialization of HWASS Shared RAM parition1 is finished. Writing '1' would clear the bit

5.2.3.31 HWASS_SHRD_RAM1_MEM_INIT_STATUS Register (Offset = 78h) [Reset = X]

HWASS_SHRD_RAM1_MEM_INIT_STATUS is shown in Table 5-252.

Return to the Summary Table.

Table 5-252. HWASS_SHRD_RAM1_MEM_INIT_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	X	
0	MEM_STATUS	R	0h	1'b 0: No initialization is happening for HWASS Shared RAM parition1 1'b 1: Initialization is in progress for HWASS Shared RAM parition1

5.2.3.32 APPSS_TPCC_MEMINIT_START Register (Offset = 7Ch) [Reset = X]

APPSS_TPCC_MEMINIT_START is shown in Table 5-253.

Return to the Summary Table.

Table 5-253. APPSS_TPCC_MEMINIT_START Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	TPCC_B_MEMINIT_START	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the TPCCB
15-1	RESERVED	R/W	X	
0	TPCC_A_MEMINIT_START	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the TPCCA

5.2.3.33 APPSS_TPCC_MEMINIT_DONE Register (Offset = 80h) [Reset = X]

APPSS_TPCC_MEMINIT_DONE is shown in Table 5-254.

Return to the Summary Table.

Table 5-254. APPSS_TPCC_MEMINIT_DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	

Table 5-254. APPSS_TPCC_MEMINIT_DONE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	TPCC_B_MEMINIT_DONE	R/W	0h	This field will be high once intialization of TPCCB is finished. Writing '1' would clear the bit
15-1	RESERVED	R/W	X	
0	TPCC_A_MEMINIT_DONE	R/W	0h	This field will be high once intialization of TPCCA is finished. Writing '1' would clear the bit

5.2.3.34 APPSS_TPCC_MEMINIT_STATUS Register (Offset = 84h) [Reset = X]

APPSS_TPCC_MEMINIT_STATUS is shown in Table 5-255.

Return to the Summary Table.

Table 5-255. APPSS_TPCC_MEMINIT_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	X	
16	TPCC_B_MEMINIT_STATUS	R	0h	1'b 0: No initialization is happening for TPCCA 1'b 1: Initialization is in progress for TPCCB
15-1	RESERVED	R	X	
0	TPCC_A_MEMINIT_STATUS	R	0h	1'b 0: No initialization is happening for TPCCA 1'b 1: Initialization is in progress for TPCCB

5.2.3.35 APPSS_SPIA_CFG Register (Offset = 88h) [Reset = X]

APPSS_SPIA_CFG is shown in Table 5-256.

Return to the Summary Table.

Table 5-256. APPSS_SPIA_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	X	
28	SPIA_IODFT_EN	R/W	0h	1: Enable loop back of MOSI to MISO - Master mode Enable loop back of MISO to MOSI - Slave mode
27-25	RESERVED	R/W	X	
24	SPIA_INT_TRIG_POLARITY	R/W	0h	SPIA trigger source polarity select. 0 - Polarity 0, 1 -Polarity 1
23-17	RESERVED	R/W	X	
16	SPIA_TRIG_GATE_EN	R/W	0h	When set the TRIGGER s are un-gated only when chip-select is active
15-9	RESERVED	R/W	X	
8	SPIA_CS_TRIGSRC_EN	R/W	0h	MIBSPIB CS Trigger SRC enable 1 : Use CS as trigger source
7-1	RESERVED	R/W	X	
0	SPIASYNC2SEN	R/W	0h	Donot touch the field. Used as Tie-off for IP-config.

5.2.3.36 APPSS_SPIB_CFG Register (Offset = 8Ch) [Reset = X]

APPSS_SPIB_CFG is shown in Table 5-257.

Return to the Summary Table.

Table 5-257. APPSS_SPIB_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	X	
28	SPIB_IODFT_EN	R/W	0h	1: Enable loop back of MOSI to MISO - Master mode Enable loop back of MISO to MOSI - Slave mode
27-25	RESERVED	R/W	X	
24	SPIB_INT_TRIG_POLARITY	R/W	0h	SPIB trigger source polarity select. 0 - Polarity 0, 1 -Polarity 1
23-17	RESERVED	R/W	X	
16	SPIB_TRIG_GATE_EN	R/W	0h	When set the TRIGGER s are un-gated only when chip-select is active
15-9	RESERVED	R/W	X	
8	SPIB_CS_TRIGSRC_EN	R/W	0h	MIBSPIB CS Trigger SRC enable 1 : Use CS as trigger source
7-1	RESERVED	R/W	X	
0	SPIBSYNC2SEN	R/W	0h	Donot touch the field. Used as Tie-off for IP-config.

5.2.3.37 APPSS_EPWM_CFG Register (Offset = 90h) [Reset = 0F000000h]

APPSS_EPWM_CFG is shown in Table 5-258.

Return to the Summary Table.

Table 5-258. APPSS_EPWM_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EPWM_CONFIG	R/W	0F000000h	bit0: SW syncin for EPWM1 bit1: SW syncin for EPWM2 bit2: SW syncin for EPWM3 bit8:9 : select bits for EPWM1 '0' : external syncin '1' : reserved '2' : sw syncin '3' : reserved bit10:11 : select bits for EPWM2 '0' : external syncin '1' : chained from EPWM1 '2' : sw syncin '3' : reserved bit12:13 : select bits for EPWM3 '0' : external syncin '1' : chained from EPWM2 '2' : sw syncin '3' : reserved bit24:TBCLKEN for EPWM1 bit25:TBCLKEN for EPWM2 bit26:TBCLKEN for EPWM3

5.2.3.38 RESERVED Register (Offset = 94h) [Reset = 00000000h]

RESERVED is shown in Table 5-259.

Return to the Summary Table.

Table 5-259. RESERVED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GIO_CONFIG	R/W	0h	bit0 : writing '1' will slect negedge for pulse generation of GIO_PAD_INT0 to IRQ bit1 : writing '1' will slect negedge for pulse generation of GIO_PAD_INT1to IRQ bit2: writing '1' will slect negedge for pulse generation of GIO_PAD_INT2 to IRQ bit3 : writing '1' will slect negedge for pulse generation of GIO_PAD_INT3 to IRQ bit4 : writing '1' will slect negedge for pulse generation of GIO_PAD_INT4 to IRQ bit5 : writing '1' will slect negedge for pulse generation of GIO_PAD_INT5 to IRQ bit6 : writing '1' will slect negedge for pulse generation of GIO_PAD_INT6 to IRQ bit7 : writing '1' will slect negedge for pulse generation of GIO_PAD_INT7 to IRQ

5.2.3.39 APPSS_MCAN_FE_AND_LIN_INTR_SEL Register (Offset = 98h) [Reset = X]

APPSS_MCAN_FE_AND_LIN_INTR_SEL is shown in Table 5-260.

Return to the Summary Table.

Table 5-260. APPSS_MCAN_FE_AND_LIN_INTR_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	LIN_INTR_SEL	R/W	0h	Writing a value would select the LIN interrupt in combination with HW_SYNC_IN and CAN filter events for Frame timer 0 : 0th interrupt bit is selected 1 : 1st interrupt bit is selected
2-0	MCAN_FE_SEL	R/W	0h	Writing a value 'N' would select Nth filter interrupt combination with SYNC_IN(IO) for triggering timing engine Example: writing 3'd< 1-7> selects MCAN_FE_INT< 1-7> respectively

5.2.3.40 APPSS_MCANA_INT_CLR Register (Offset = 9Ch) [Reset = 00000000h]

APPSS_MCANA_INT_CLR is shown in Table 5-261.

Return to the Summary Table.

Table 5-261. APPSS_MCANA_INT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MCAN_INT_CLR	R/W	0h	Interrupt Clear for 32 MCANSS TX DMA interrupts. Writing 1'b1 to bit<0-31> clears interrupt source <0-31> respectively in MCANA

5.2.3.41 APPSS_MCANA_INT_MASK Register (Offset = A0h) [Reset = 00000000h]

APPSS_MCANA_INT_MASK is shown in Table 5-262.

Return to the Summary Table.

Table 5-262. APPSS_MCANA_INT_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MCAN_INT_MASK	R/W	0h	Interrupt Mask for 32 MCANSS TX DMA interrupts. Writing 1'b1 to bit<0-31> masks interrupt source <0-31> respectively in MCANA

5.2.3.42 APPSS_MCANA_INT_STAT Register (Offset = A4h) [Reset = 00000000h]

APPSS_MCANA_INT_STAT is shown in Table 5-263.

Return to the Summary Table.

Table 5-263. APPSS_MCANA_INT_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MCAN_INT_STATUS	R	0h	Interrupt status for 32 MCANSS TX DMA interrupts. 1'b1 in bit<0-31> gives pending status for interrupt <0-31> respectively in MCANA

5.2.3.43 APPSS_CM4_GLOBAL_CONFIG Register (Offset = A8h) [Reset = X]

APPSS_CM4_GLOBAL_CONFIG is shown in Table 5-264.

Return to the Summary Table.

Table 5-264. APPSS_CM4_GLOBAL_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	

Table 5-264. APPSS_CM4_GLOBAL_CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	TEINIT	R/W	0h	Reserved

5.2.3.44 RESERVED1 Register (Offset = ACh) [Reset = 00000000h]

RESERVED1 is shown in Table 5-265.

Return to the Summary Table.

Table 5-265. RESERVED1 Register Field Descriptions

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

5.2.3.45 APPSS_AHB_CTRL Register (Offset = B8h) [Reset = X]

APPSS_AHB_CTRL is shown in Table 5-266.

Return to the Summary Table.

Table 5-266. APPSS_AHB_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	CPU0_AHB_INIT	R/W	1h	Ti internal Register. Modifying this register is not recommended Signal decides whehter ahb interface is enabled or not.

5.2.3.46 ESM_GATING0 Register (Offset = BCh) [Reset = FFFFFFFFh]

ESM_GATING0 is shown in Table 5-267.

Return to the Summary Table.

Table 5-267. ESM_GATING0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_0 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_1 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_7

5.2.3.47 ESM_GATING1 Register (Offset = C0h) [Reset = FFFFFFFFh]

ESM_GATING1 is shown in Table 5-268.

Return to the Summary Table.

Table 5-268. ESM_GATING1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_8 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_9 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_15

5.2.3.48 ESM_GATING2 Register (Offset = C4h) [Reset = FFFFFFFFh]

ESM_GATING2 is shown in Table 5-269.

Return to the Summary Table.

Table 5-269. ESM_GATING2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_16 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_17 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_23

5.2.3.49 ESM_GATING3 Register (Offset = C8h) [Reset = FFFFFFFFh]

ESM_GATING3 is shown in Table 5-270.

Return to the Summary Table.

Table 5-270. ESM_GATING3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_24 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_25 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_31

5.2.3.50 ESM_GATING4 Register (Offset = CCh) [Reset = FFFFFFFFh]

ESM_GATING4 is shown in Table 5-271.

Return to the Summary Table.

Table 5-271. ESM_GATING4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_0 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_1 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_7

5.2.3.51 ESM_GATING5 Register (Offset = D0h) [Reset = FFFFFFFFh]

ESM_GATING5 is shown in Table 5-272.

Return to the Summary Table.

Table 5-272. ESM_GATING5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_8 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_9 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_15

5.2.3.52 ESM_GATING6 Register (Offset = D4h) [Reset = FFFFFFFFh]

ESM_GATING6 is shown in Table 5-273.

Return to the Summary Table.

Table 5-273. ESM_GATING6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_16 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_17 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_23

5.2.3.53 ESM_GATING7 Register (Offset = D8h) [Reset = FFFFFFFFh]

ESM_GATING7 is shown in Table 5-274.

Return to the Summary Table.

Table 5-274. ESM_GATING7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ESM_GATING	R/W	FFFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_24 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_25 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_31

5.2.3.54 APPSS_CM4_HALT Register (Offset = DCh) [Reset = X]

APPSS_CM4_HALT is shown in Table 5-275.

Return to the Summary Table.

Table 5-275. APPSS_CM4_HALT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	HALT	R/W	7h	RESERVED

5.2.3.55 APPSS_CM4_EVENT Register (Offset = E0h) [Reset = X]

APPSS_CM4_EVENT is shown in Table 5-276.

Return to the Summary Table.

Table 5-276. APPSS_CM4_EVENT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	CPU0_EVENT	R/W	0h	Reserved Register for R & D

5.2.3.56 SPIA_IO_CFG Register (Offset = E4h) [Reset = X]

SPIA_IO_CFG is shown in Table 5-277.

Return to the Summary Table.

Table 5-277. SPIA_IO_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	MISO_OEN_BY_CS	R/W	0h	RESERVED
15-9	RESERVED	R/W	X	
8	CS_POL	R/W	0h	RESERVED
7-1	RESERVED	R/W	X	
0	CS_DEACT	R/W	0h	RESERVED

5.2.3.57 SPIB_IO_CFG Register (Offset = E8h) [Reset = X]

SPIB_IO_CFG is shown in Table 5-278.

Return to the Summary Table.

Table 5-278. SPIB_IO_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	MISO_OEN_BY_CS	R/W	0h	RESERVED
15-9	RESERVED	R/W	X	

Table 5-278. SPIB_IO_CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	CS_POL	R/W	0h	RESERVED
7-1	RESERVED	R/W	X	
0	CS_DEACT	R/W	0h	RESERVED

5.2.3.58 SPI_HOST_IRQ Register (Offset = ECh) [Reset = X]

SPI_HOST_IRQ is shown in Table 5-279.

Return to the Summary Table.

Table 5-279. SPI_HOST_IRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	HOST_IRQ	R/W	0h	RESERVED

5.2.3.59 TPTC_DBS_CONFIG Register (Offset = F0h) [Reset = X]

TPTC_DBS_CONFIG is shown in Table 5-280.

Return to the Summary Table.

Table 5-280. TPTC_DBS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R/W	X	
13-12	TPTC_B1	R/W	1h	Default burst size tieoff value for TPTC_B1
11-10	RESERVED	R/W	X	
9-8	TPTC_B0	R/W	1h	Default burst size tieoff value for TPTC_B0
7-6	RESERVED	R/W	X	
5-4	TPTC_A1	R/W	1h	Default burst size tieoff value for TPTC_A1
3-2	RESERVED	R/W	X	
1-0	TPTC_A0	R/W	2h	Default burst size tieoff value for TPTC_A0

5.2.3.60 TPCC_PARITY_CTRL Register (Offset = F4h) [Reset = X]

TPCC_PARITY_CTRL is shown in Table 5-281.

Return to the Summary Table.

Table 5-281. TPCC_PARITY_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	X	
20	TPCC_B_PARITY_ERR_CLR	R/W	0h	Write pulse bit field: parity clear bit. Writing 1'b1 will clear the tpcc_b_parity_addr
19-17	RESERVED	R/W	X	
16	TPCC_A_PARITY_ERR_CLR	R/W	0h	Write pulse bit field: parity clear bit. Writing 1'b1 will clear the tpcc_a_parity_addr
15-13	RESERVED	R/W	X	
12	TPCC_B_PARITY_TESTEN	R/W	0h	parity test enable for tpcc b
11-9	RESERVED	R/W	X	
8	TPCC_B_PARITY_EN	R/W	0h	parity en for tpcc b
7-5	RESERVED	R/W	X	

Table 5-281. TPCC_PARITY_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TPCC_A_PARITY_TESTEN	R/W	0h	parity test enable for tpcc a
3-1	RESERVED	R/W	X	
0	TPCC_A_PARITY_EN	R/W	0h	writing 1'b1 enables parity for TPCC_A

5.2.3.61 TPCC_PARITY_STATUS Register (Offset = F8h) [Reset = X]

TPCC_PARITY_STATUS is shown in Table 5-282.

Return to the Summary Table.

Table 5-282. TPCC_PARITY_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	X	
23-16	TPCC_B_PARITY_ADDR	R	0h	address where parity error happened for tpccb
15-8	RESERVED	R	X	
7-0	TPCC_A_PARITY_ADDR	R	0h	address where parity error happened for tpcca

5.2.3.62 APPSS_DBG_ACK_CTL0 Register (Offset = FCh) [Reset = X]

APPSS_DBG_ACK_CTL0 is shown in Table 5-283.

Return to the Summary Table.

Table 5-283. APPSS_DBG_ACK_CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	X	
28	LIN	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
27-25	RESERVED	R/W	X	
24	SCIB	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
23-21	RESERVED	R/W	X	
20	SCIA	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
19-17	RESERVED	R/W	X	
16	I2C	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
15-13	RESERVED	R/W	X	
12	MCRC	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
11-9	RESERVED	R/W	X	
8	WDT	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
7-5	RESERVED	R/W	X	
4	RTI	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor

Table 5-283. APPSS_DBG_ACK_CTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-1	RESERVED	R/W	X	
0	CAN	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor

5.2.3.63 DEBUGSS_CSETB_FLUSH Register (Offset = 100h) [Reset = X]

DEBUGSS_CSETB_FLUSH is shown in Table 5-284.

Return to the Summary Table.

Table 5-284. DEBUGSS_CSETB_FLUSH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	X	
10	CSETB_FULL	R	0h	RESERVED
9	CSETB_ACQ_COMPLETE	R	0h	RESERVED
8	CSETB_FLUSHINACK	R	0h	RESERVED
7-1	RESERVED	R/W	X	
0	CSETB_FLUSHIN	R/W	0h	RESERVED

5.2.3.64 CPSW_CONTROL Register (Offset = 104h) [Reset = X]

CPSW_CONTROL is shown in Table 5-285.

Return to the Summary Table.

Table 5-285. CPSW_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	RGMII1_ID_MODE	R/W	0h	Reserved
15-9	RESERVED	R/W	X	
8	RMII_REF_CLK_OE_N	R/W	0h	Reserved
7-1	RESERVED	R/W	X	
0	PORT1_MODE_SEL	R/W	0h	Reserved

5.2.3.65 APPSS_ERRAGG_MASK0 Register (Offset = 108h) [Reset = X]

APPSS_ERRAGG_MASK0 is shown in Table 5-286.

Return to the Summary Table.

Table 5-286. APPSS_ERRAGG_MASK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	X	
25	APP_AHB_SLV_RD	R/W	1h	Mask Interrupt from AHB slaves to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
24	APP_AHB_SLV_WR	R/W	1h	Mask Interrupt from AHB slaves to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

Table 5-286. APPSS_ERRAGG_MASK0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	FEC_ERRORAGG	R/W	0h	Mask Interrupt from FEC_ERRORAGG to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
22	APP_SHARED_MEM	R/W	0h	Mask Interrupt from APP_SHARED_MEM to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
21	APP_AHB	R/W	0h	Mask Interrupt from APP_AHB to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
20	APP_MPU	R/W	0h	Mask Interrupt from APP_MPU to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
19	TOP_EFUSE_CTRL_WR	R/W	0h	Mask Interrupt from TOP_EFUSE_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
18	TOP_EFUSE_CTRL_RD	R/W	0h	Mask Interrupt from FEC_ERRORAGG to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
17	TOP_PRCM_WR	R/W	0h	Mask Interrupt from TOP_PRCM to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
16	TOP_PRCM_RD	R/W	0h	Mask Interrupt from TOP_PRCM to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15	FRAME_TIMER_WR	R/W	0h	Mask Interrupt from FRAME_TIMER to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
14	FRAME_TIMER_RD	R/W	0h	Mask Interrupt from FRAME_TIMER to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
13	APLL_CTRL_WR	R/W	0h	Mask Interrupt from APLL_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
12	APLL_CTRL_RD	R/W	0h	Mask Interrupt from APLL_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
11	TOPSS_CTRL_WR	R/W	0h	Mask Interrupt from TOPSS_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
10	TOPSS_CTRL_RD	R/W	0h	Mask Interrupt from TOPSS_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
9	PLLDIG_CTRL_WR	R/W	0h	Mask Interrupt from PLLDIG_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

Table 5-286. APPSS_ERRAGG_MASK0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PLLDIG_CTRL_RD	R/W	0h	Mask Interrupt from PLLDIG_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7	ADCBUFF_CTRL_WR	R/W	0h	Mask Interrupt from ADCBUFF_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	ADCBUFF_CTRL_RD	R/W	0h	Mask Interrupt from ADCBUFF_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	APP_IDALLOC_WR	R/W	0h	Mask Interrupt from APP_IDALLOC to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	APP_IDALLOC_RD	R/W	0h	Mask Interrupt from APP_IDALLOC to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
3	APP_CTRL_WR	R/W	0h	Mask Interrupt from APP_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	APP_CTRL_RD	R/W	0h	Mask Interrupt from APP_CTRL to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	APP_RCM_WR	R/W	0h	Mask Interrupt from APP_RCM to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	APP_RCM_RD	R/W	0h	Mask Interrupt from APP_RCM to aggregated Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

ADVANCE INFORMATION

5.2.3.66 APPSS_ERRAGG_STATUS0 Register (Offset = 10Ch) [Reset = X]

APPSS_ERRAGG_STATUS0 is shown in Table 5-287.

Return to the Summary Table.

Table 5-287. APPSS_ERRAGG_STATUS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	X	
25	APP_AHB_SLV_RD	R/W	0h	Status of Interrupt from AHB_SLAVE Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
24	APP_AHB_SLV_WR	R/W	0h	Status of Interrupt from AHB_SLAVE Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
23	FEC_ERRORAGG	R/W	0h	Status of Interrupt from FEC_ERRORAGG Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
22	APP_SHARED_MEM_ERR	R/W	0h	Status of Interrupt from APP_SHARED_MEM Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
21	APP_AHB_WR	R/W	0h	Status of Interrupt from APP_AHB Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.

Table 5-287. APPSS_ERRAGG_STATUS0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	APP_MPU_RD	R/W	0h	Status of Interrupt from APP_MPU Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
19	TOP_EFUSE_CTRL_WR	R/W	0h	Status of Interrupt from TOP_EFUSE_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
18	TOP_EFUSE_CTRL_RD	R/W	0h	Status of Interrupt from TOP_EFUSE_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
17	TOP_PRCM_WR	R/W	0h	Status of Interrupt from TOP_PRCM Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
16	TOP_PRCM_RD	R/W	0h	Status of Interrupt from TOP_PRCM Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
15	FRAME_TIMER_WR	R/W	0h	Status of Interrupt from FRAME_TIMER Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
14	FRAME_TIMER_RD	R/W	0h	Status of Interrupt from FRAME_TIMER Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
13	APLL_CTRL_WR	R/W	0h	Status of Interrupt from APLL_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
12	APLL_CTRL_RD	R/W	0h	Status of Interrupt from APLL_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
11	TOPSS_CTRL_WR	R/W	0h	Status of Interrupt from TOPSS_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
10	TOPSS_CTRL_RD	R/W	0h	Status of Interrupt from TOPSS_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
9	PLLDIG_CTRL_WR	R/W	0h	Status of Interrupt from PLLDIG_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
8	PLLDIG_CTRL_RD	R/W	0h	Status of Interrupt from PLLDIG_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
7	ADCBUFF_CTRL_WR	R/W	0h	Status of Interrupt from ADCBUFF_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
6	ADCBUFF_CTRL_RD	R/W	0h	Status of Interrupt from ADCBUFF_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
5	APP_IDALLOC_WR	R/W	0h	Status of Interrupt from APP_IDALLOC Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
4	APP_IDALLOC_RD	R/W	0h	Status of Interrupt from APP_IDALLOC Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
3	APP_CTRL_WR	R/W	0h	Status of Interrupt from APP_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
2	APP_CTRL_RD	R/W	0h	Status of Interrupt from APP_CTRL Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
1	APP_RCM_WR	R/W	0h	Status of Interrupt from APP_RCM Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
0	APP_RCM_RD	R/W	0h	Status of Interrupt from APP_RCM Set only if Interupt is unmasked in APPSS_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.

5.2.3.67 APPSS_TPCC_A_ERRAGG_MASK Register (Offset = 190h) [Reset = X]

APPSS_TPCC_A_ERRAGG_MASK is shown in Table 5-288.

Return to the Summary Table.

Table 5-288. APPSS_TPCC_A_ERRAGG_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26	TPTC_A1_READ_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_A1 to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
25	TPTC_A0_READ_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_A0 to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
24	TPCC_A_READ_ACCESS_ERROR	R/W	0h	Mask Error from TPCC_A to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
23-19	RESERVED	R/W	X	
18	TPTC_A1_WRITE_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_A1 to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
17	TPTC_A0_WRITE_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_A0 to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
16	TPCC_A_WRITE_ACCESS_ERROR	R/W	0h	Mask Error from TPCC_A to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
15-5	RESERVED	R/W	X	
4	TPCC_A_PAR_ERR	R/W	0h	Mask Error from TPCC_A to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
3	TPTC_A1_ERR	R/W	0h	Mask Error from TPTC_A1 to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
2	TPTC_A0_ERR	R/W	0h	Mask Error from TPTC_A0 to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
1	TPCC_A_MPINT	R/W	0h	Mask Error from TPCC_A to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
0	TPCC_A_ERRINT	R/W	0h	Mask Error from TPCC_A to aggregated Error TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked

5.2.3.68 APPSS_TPCC_A_ERRAGG_STATUS Register (Offset = 194h) [Reset = X]

APPSS_TPCC_A_ERRAGG_STATUS is shown in Table 5-289.

Return to the Summary Table.

Table 5-289. APPSS_TPCC_A_ERRAGG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26	TPTC_A1_READ_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_A1. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
25	TPTC_A0_READ_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_A0. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.

Table 5-289. APPSS_TPCC_A_ERRAGG_STATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	TPCC_A_READ_ACCESS_ERROR	R/W	0h	Status of Error from TPCC_A. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
23-19	RESERVED	R/W	X	
18	TPTC_A1_WRITE_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_A1. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
17	TPTC_A0_WRITE_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_A0. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
16	TPCC_A_WRITE_ACCESS_ERROR	R/W	0h	Status of Error from TPCC_A. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
15-5	RESERVED	R/W	X	
4	TPCC_A_PAR_ERR	R/W	0h	Status of Error from TPCC_A. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
3	TPTC_A1_ERR	R/W	0h	Status of Error from TPTC_A1. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
2	TPTC_A0_ERR	R/W	0h	Status of Error from TPTC_A0. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
1	TPCC_A_MPINT	R/W	0h	Status of Error from TPCC_A. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
0	TPCC_A_ERRINT	R/W	0h	Status of Error from TPCC_A. Set only if Interupt is unmasked in TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.

5.2.3.69 APPSS_TPCC_A_ERRAGG_STATUS_RAW Register (Offset = 198h) [Reset = X]

APPSS_TPCC_A_ERRAGG_STATUS_RAW is shown in Table 5-290.

Return to the Summary Table.

Table 5-290. APPSS_TPCC_A_ERRAGG_STATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26	TPTC_A1_READ_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_A1. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
25	TPTC_A0_READ_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_A0. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
24	TPCC_A_READ_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
23-19	RESERVED	R/W	X	
18	TPTC_A1_WRITE_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_A1. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
17	TPTC_A0_WRITE_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_A0. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK

Table 5-290. APPSS_TPCC_A_ERRAGG_STATUS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	TPCC_A_WRITE_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
15-5	RESERVED	R/W	X	
4	TPCC_A_PAR_ERR	R/W	0h	Raw Status of Error from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
3	TPTC_A1_ERR	R/W	0h	Raw Status of Error from TPTC_A1. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
2	TPTC_A0_ERR	R/W	0h	Raw Status of Error from TPTC_A0. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
1	TPCC_A_MPINT	R/W	0h	Raw Status of Error from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK
0	TPCC_A_ERRINT	R/W	0h	Raw Status of Error from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_A_ERRAGG_MASK

5.2.3.70 APPSS_TPCC_A_INTAGG_MASK Register (Offset = 214h) [Reset = X]

APPSS_TPCC_A_INTAGG_MASK is shown in Table 5-291.

Return to the Summary Table.

Table 5-291. APPSS_TPCC_A_INTAGG_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	TPTC_A1	R/W	0h	Mask Interrupt from TPTC A1 to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
16	TPTC_A0	R/W	0h	Mask Interrupt from TPTC A0 to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15-9	RESERVED	R/W	X	
8	TPCC_A_INT7	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7	TPCC_A_INT6	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	TPCC_A_INT5	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	TPCC_A_INT4	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	TPCC_A_INT3	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

Table 5-291. APPSS_TPCC_A_INTAGG_MASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	TPCC_A_INT2	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	TPCC_A_INT1	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	TPCC_A_INT0	R/W	0h	Mask Interrupt from TPCC A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	TPCC_A_INTG	R/W	0h	Mask Interrupt from TPCC_A to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

ADVANCE INFORMATION

5.2.3.71 HW_SPARE_WPH Register (Offset = 218h) [Reset = X]

HW_SPARE_WPH is shown in Table 5-292.

Return to the Summary Table.

Table 5-292. HW_SPARE_WPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	TPTC_A1	R/W	0h	Status of Interrupt from TPTC A1. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
16	TPTC_A0	R/W	0h	Status of Interrupt from TPTC A0. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
15-9	RESERVED	R/W	X	
8	TPCC_A_INT7	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
7	TPCC_A_INT6	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
6	TPCC_A_INT5	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
5	TPCC_A_INT4	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
4	TPCC_A_INT3	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
3	TPCC_A_INT2	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
2	TPCC_A_INT1	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
1	TPCC_A_INT0	R/W	0h	Status of Interrupt from TPCC A Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.

Table 5-292. HW_SPARE_WPH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	TPCC_A_INTG	R/W	0h	Status of Interrupt from TPCC_A. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.

5.2.3.72 APPSS_TPCC_A_INTAGG_STATUS_RAW Register (Offset = 21Ch) [Reset = X]

APPSS_TPCC_A_INTAGG_STATUS_RAW is shown in Table 5-293.

Return to the Summary Table.

Table 5-293. APPSS_TPCC_A_INTAGG_STATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	TPTC_A1	R/W	0h	Raw Status of Interrupt from TPTC A1. Set irrespective if the Interupt is masked or unmasked in TPCC_A_INTAGG_MASK
16	TPTC_A0	R/W	0h	Raw Status of Interrupt from TPTC A0. Set irrespective if the Interupt is masked or unmasked in TPCC_A_INTAGG_MASK
15-9	RESERVED	R/W	X	
8	TPCC_A_INT7	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
7	TPCC_A_INT6	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
6	TPCC_A_INT5	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
5	TPCC_A_INT4	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
4	TPCC_A_INT3	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
3	TPCC_A_INT2	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
2	TPCC_A_INT1	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
1	TPCC_A_INT0	R/W	0h	Raw Status of Interrupt from TPCC A. Set irrespective if the Interupt is masked or unmasked in TPCC_A_INTAGG_MASK
0	TPCC_A_INTG	R/W	0h	Raw Status of Interrupt from TPCC_A. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK

5.2.3.73 APPSS_TPCC_B_ERRAGG_MASK Register (Offset = 274h) [Reset = X]

APPSS_TPCC_B_ERRAGG_MASK is shown in Table 5-294.

Return to the Summary Table.

Table 5-294. APPSS_TPCC_B_ERRAGG_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	

Table 5-294. APPSS_TPCC_B_ERRAGG_MASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	TPTC_B1_READ_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_B0 to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
25	TPTC_B0_READ_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_B0 to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
24	TPCC_B_READ_ACCESS_ERROR	R/W	0h	Mask Error from TPCC_B to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
23-18	RESERVED	R/W	X	
17	TPTC_B0_WRITE_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_B0 to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
16	TPCC_B_WRITE_ACCESS_ERROR	R/W	0h	Mask Error from TPCC_B to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
15	RESERVED	R/W	X	
14	TPTC_B1_WRITE_ACCESS_ERROR	R/W	0h	Mask Error from TPTC_B0 to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
13-5	RESERVED	R/W	X	
4	TPCC_B_PAR_ERR	R/W	0h	Mask Error from TPCC_B to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
3	TPTC_B1_ERR	R/W	0h	Mask Error from TPTC_B0 to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
2	TPTC_B0_ERR	R/W	0h	Mask Error from TPTC_B0 to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
1	TPCC_B_MPINT	R/W	0h	Mask Error from TPCC_B to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
0	TPCC_B_ERRINT	R/W	0h	Mask Error from TPCC_B to aggregated Error TPCC_B_ERRAGG 1 : Error is Masked 0 : Error is Unmasked

5.2.3.74 APPSS_TPCC_B_ERRAGG_STATUS Register (Offset = 278h) [Reset = X]

APPSS_TPCC_B_ERRAGG_STATUS is shown in Table 5-295.

Return to the Summary Table.

Table 5-295. APPSS_TPCC_B_ERRAGG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26	TPTC_B1_READ_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_B0. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
25	TPTC_B0_READ_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_B0. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
24	TPCC_B_READ_ACCESS_ERROR	R/W	0h	Status of Error from TPCC_B. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
23-18	RESERVED	R/W	X	

Table 5-295. APPSS_TPCC_B_ERRAGG_STATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	TPTC_B0_WRITE_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_B0. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
16	TPCC_B_WRITE_ACCESS_ERROR	R/W	0h	Status of Error from TPCC_B. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
15	RESERVED	R/W	X	
14	TPTC_B1_WRITE_ACCESS_ERROR	R/W	0h	Status of Error from TPTC_B0. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
13-5	RESERVED	R/W	X	
4	TPCC_B_PAR_ERR	R/W	0h	Status of Error from TPCC_B. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
3	TPTC_B1_ERR	R/W	0h	Status of Error from TPTC_B0. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
2	TPTC_B0_ERR	R/W	0h	Status of Error from TPTC_B0. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
1	TPCC_B_MPINT	R/W	0h	Status of Error from TPCC_B. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.
0	TPCC_B_ERRINT	R/W	0h	Status of Error from TPCC_B. Set only if Interupt is unmasked in TPCC_B_ERRAGG_MASK Wrie 0x1 to clear this Error.

5.2.3.75 APPSS_TPCC_B_ERRAGG_STATUS_RAW Register (Offset = 27Ch) [Reset = X]

APPSS_TPCC_B_ERRAGG_STATUS_RAW is shown in Table 5-296.

Return to the Summary Table.

Table 5-296. APPSS_TPCC_B_ERRAGG_STATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	X	
26	TPTC_B1_READ_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_B0. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
25	TPTC_B0_READ_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_B0. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
24	TPCC_B_READ_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
23-18	RESERVED	R/W	X	
17	TPTC_B0_WRITE_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_B0. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
16	TPCC_B_WRITE_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
15	RESERVED	R/W	X	
14	TPTC_B1_WRITE_ACCESS_ERROR	R/W	0h	Raw Status of Error from TPTC_B0. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
13-5	RESERVED	R/W	X	

Table 5-296. APPSS_TPCC_B_ERRAGG_STATUS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TPCC_B_PAR_ERR	R/W	0h	Raw Status of Error from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
3	TPTC_B1_ERR	R/W	0h	Raw Status of Error from TPTC_B0. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
2	TPTC_B0_ERR	R/W	0h	Raw Status of Error from TPTC_B0. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
1	TPCC_B_MPINT	R/W	0h	Raw Status of Error from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK
0	TPCC_B_ERRINT	R/W	0h	Raw Status of Error from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_B_ERRAGG_MASK

ADVANCE INFORMATION

5.2.3.76 APPSS_TPCC_B_INTAGG_MASK Register (Offset = 2ECh) [Reset = X]

APPSS_TPCC_B_INTAGG_MASK is shown in Table 5-297.

Return to the Summary Table.

Table 5-297. APPSS_TPCC_B_INTAGG_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	TPTC_B1	R/W	0h	Mask Interrupt from TPTC A0 to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
16	TPTC_B0	R/W	0h	Mask Interrupt from TPTC A0 to aggregated Interrupt TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15-9	RESERVED	R/W	X	
8	TPCC_B_INT7	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7	TPCC_B_INT6	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	TPCC_B_INT5	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	TPCC_B_INT4	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	TPCC_B_INT3	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
3	TPCC_B_INT2	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

Table 5-297. APPSS_TPCC_B_INTAGG_MASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	TPCC_B_INT1	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	TPCC_B_INT0	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	TPCC_B_INTG	R/W	0h	Mask Interrupt from TPCC_B to aggregated Interrupt TPCC_B_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

5.2.3.77 APPSS_TPCC_B_INTAGG_STATUS Register (Offset = 2F0h) [Reset = X]

APPSS_TPCC_B_INTAGG_STATUS is shown in Table 5-298.

Return to the Summary Table.

Table 5-298. APPSS_TPCC_B_INTAGG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	TPTC_B1	R/W	0h	Status of Interrupt from TPTC A0. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
16	TPTC_B0	R/W	0h	Status of Interrupt from TPTC A0. Set only if Interupt is unmasked in TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
15-9	RESERVED	R/W	X	
8	TPCC_B_INT7	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
7	TPCC_B_INT6	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
6	TPCC_B_INT5	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
5	TPCC_B_INT4	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
4	TPCC_B_INT3	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
3	TPCC_B_INT2	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
2	TPCC_B_INT1	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
1	TPCC_B_INT0	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.
0	TPCC_B_INTG	R/W	0h	Status of Interrupt from TPCC_B. Set only if Interupt is unmasked in TPCC_B_INTAGG_MASK Wrie 0x1 to clear this interrupt.

5.2.3.78 APPSS_TPCC_B_INTAGG_STATUS_RAW Register (Offset = 2F4h) [Reset = X]

APPSS_TPCC_B_INTAGG_STATUS_RAW is shown in Table 5-299.

Return to the Summary Table.

Table 5-299. APPSS_TPCC_B_INTAGG_STATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	X	
17	TPTC_B1	R/W	0h	Raw Status of Interrupt from TPTC A0. Set irrespective if the Interupt is masked or unmasked in TPCC_A_INTAGG_MASK
16	TPTC_B0	R/W	0h	Raw Status of Interrupt from TPTC A0. Set irrespective if the Interupt is masked or unmasked in TPCC_A_INTAGG_MASK
15-9	RESERVED	R/W	X	
8	TPCC_B_INT7	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
7	TPCC_B_INT6	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
6	TPCC_B_INT5	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
5	TPCC_B_INT4	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
4	TPCC_B_INT3	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
3	TPCC_B_INT2	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
2	TPCC_B_INT1	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
1	TPCC_B_INT0	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK
0	TPCC_B_INTG	R/W	0h	Raw Status of Interrupt from TPCC_B. Set irrespective if the Interupt is masked or unmasked in TPCC_C_INTAGG_MASK

5.2.3.79 APPSS_MPU_ERRAGG_MASK Register (Offset = 2F8h) [Reset = X]

APPSS_MPU_ERRAGG_MASK is shown in Table 5-300.

Return to the Summary Table.

Table 5-300. APPSS_MPU_ERRAGG_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	FECSS_MPU	R/W	0h	Mask Interrupt from FECSS MPU to aggregated Interrupt MPU_PROT_AGG_ERR 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15-1	RESERVED	R/W	X	

Table 5-300. APPSS_MPU_ERRAGG_MASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	APPSS_MPU	R/W	0h	Mask Interrupt from APSS MPU to aggregated Interrupt MPU_PROT_AGG_ERR 1 : Interrupt is Masked 0 : Interrupt is Unmasked

5.2.3.80 APPSS_MPU_ERRAGG_STATUS Register (Offset = 2FCh) [Reset = X]

APPSS_MPU_ERRAGG_STATUS is shown in Table 5-301.

Return to the Summary Table.

Table 5-301. APPSS_MPU_ERRAGG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	FECSS_MPU	R/W	0h	Status of Interrupt from FECSS MPU. Set only if Interupt is unmasked in APPSS_MPU_ERRAGG_MASK Wrie 0x1 to clear this interrupt.
15-1	RESERVED	R/W	X	
0	APPSS_MPU	R/W	0h	Status of Interrupt from APSS MPU. Set only if Interupt is unmasked in APPSS_MPU_ERRAGG_MASK Wrie 0x1 to clear this interrupt.

5.2.3.81 APPSS_MPU_ERRAGG_STATUS_RAW Register (Offset = 300h) [Reset = X]

APPSS_MPU_ERRAGG_STATUS_RAW is shown in Table 5-302.

Return to the Summary Table.

Table 5-302. APPSS_MPU_ERRAGG_STATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	FECSS_MPU	R/W	0h	Raw Status of FECSS MPU PROT ERR. Set irrespective if the Interupt is masked or unmasked in APPSS_MPU_ERRAGG_MASK
15-1	RESERVED	R/W	X	
0	APPSS_MPU	R/W	0h	Raw Status of Interrupt from APSS MPU PROT ERR Set irrespective if the Interupt is masked or unmasked in APPSS_MPU_ERRAGG_MASK

5.2.3.82 APPSS_QSPI_CONFIG Register (Offset = 304h) [Reset = X]

APPSS_QSPI_CONFIG is shown in Table 5-303.

Return to the Summary Table.

Table 5-303. APPSS_QSPI_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	CLK_LOOPBACK	R/W	0h	Reserved
7-1	RESERVED	R/W	X	
0	EXT_CLK	R/W	0h	Reserved

5.2.3.83 APPSS_CTI_TRIG_SEL Register (Offset = 308h) [Reset = X]

APPSS_CTI_TRIG_SEL is shown in Table 5-304.

Return to the Summary Table.

Table 5-304. APPSS_CTI_TRIG_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	TRIG8_SEL	R/W	0h	Used for selecting the trigger source for 8th trigger of CTI

5.2.3.84 APPSS_DBGSS_CTI_TRIG_SEL Register (Offset = 30Ch) [Reset = X]

APPSS_DBGSS_CTI_TRIG_SEL is shown in Table 5-305.

Return to the Summary Table.

Table 5-305. APPSS_DBGSS_CTI_TRIG_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	X	
23-16	TRIG3	R/W	0h	Reserved
15-8	TRIG2	R/W	0h	Reserved
7-0	TRIG1	R/W	0h	Reserved

5.2.3.85 APPSS_BOOT_INFO_REG0 Register (Offset = 310h) [Reset = 00000000h]

APPSS_BOOT_INFO_REG0 is shown in Table 5-306.

Return to the Summary Table.

Table 5-306. APPSS_BOOT_INFO_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.86 APPSS_BOOT_INFO_REG1 Register (Offset = 314h) [Reset = 00000000h]

APPSS_BOOT_INFO_REG1 is shown in Table 5-307.

Return to the Summary Table.

Table 5-307. APPSS_BOOT_INFO_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.87 APPSS_BOOT_INFO_REG2 Register (Offset = 318h) [Reset = 00000000h]

APPSS_BOOT_INFO_REG2 is shown in Table 5-308.

Return to the Summary Table.

Table 5-308. APPSS_BOOT_INFO_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.88 APPSS_BOOT_INFO_REG3 Register (Offset = 31Ch) [Reset = 00000000h]

APPSS_BOOT_INFO_REG3 is shown in Table 5-309.

Return to the Summary Table.

Table 5-309. APPSS_BOOT_INFO_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.89 APPSS_BOOT_INFO_REG4 Register (Offset = 320h) [Reset = 00000000h]

APPSS_BOOT_INFO_REG4 is shown in Table 5-310.

Return to the Summary Table.

Table 5-310. APPSS_BOOT_INFO_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.90 APPSS_BOOT_INFO_REG5 Register (Offset = 324h) [Reset = 00000000h]

APPSS_BOOT_INFO_REG5 is shown in Table 5-311.

Return to the Summary Table.

Table 5-311. APPSS_BOOT_INFO_REG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.91 APPSS_BOOT_INFO_REG6 Register (Offset = 328h) [Reset = 00000000h]

APPSS_BOOT_INFO_REG6 is shown in Table 5-312.

Return to the Summary Table.

Table 5-312. APPSS_BOOT_INFO_REG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.92 APPSS_BOOT_INFO_REG7 Register (Offset = 32Ch) [Reset = 00000000h]

APPSS_BOOT_INFO_REG7 is shown in Table 5-313.

Return to the Summary Table.

Table 5-313. APPSS_BOOT_INFO_REG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R/W	0h	Reserved Register for Software use

5.2.3.93 APPSS_TPTC_ECCAGGR_CLK_CNTRL Register (Offset = 330h) [Reset = X]

APPSS_TPTC_ECCAGGR_CLK_CNTRL is shown in Table 5-314.

Return to the Summary Table.

Table 5-314. APPSS_TPTC_ECCAGGR_CLK_CNTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2	TPTC_B0	R/W	1h	Writing '0' will gate the clock to TPTC_B 0-FIFO during ECC-AGGR interaction(fault injection)

Table 5-314. APPSS_TPTC_ECCAGGR_CLK_CNTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	TPTC_A1	R/W	1h	Writing '0' will gate the clock to TPTC_A 1-FIFO during ECC-AGGR interaction(fault injection)
0	TPTC_A0	R/W	1h	Writing '0' will gate the clock to TPTC_A 0-FIFO during ECC-AGGR interaction(fault injection)

5.2.3.94 APPSS_TPTC_BOUNDARY_CFG Register (Offset = 334h) [Reset = X]

APPSS_TPTC_BOUNDARY_CFG is shown in Table 5-315.

Return to the Summary Table.

Table 5-315. APPSS_TPTC_BOUNDARY_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-24	TPTC_B1_SIZE	R/W	13h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of TPTC_B1 Example: writing 6'd19 decidies boundary to be 2^{19} i.e. 512 KB
23-22	RESERVED	R/W	X	
21-16	TPTC_B0_SIZE	R/W	13h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of TPTC_B0 Example: writing 6'd19 decidies boundary to be 2^{19} i.e. 512 KB
15-14	RESERVED	R/W	X	
13-8	TPTC_A1_SIZE	R/W	13h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of TPTC_A1 Example: writing 6'd19 decidies boundary to be 2^{19} i.e. 512 KB
7-6	RESERVED	R/W	X	
5-0	TPTC_A0_SIZE	R/W	13h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of TPTC_A0 Example: writing 6'd19 decidies boundary to be 2^{19} i.e. 512 KB

5.2.3.95 APPSS_TPTC_XID_REORDER_CFG Register (Offset = 338h) [Reset = X]

APPSS_TPTC_XID_REORDER_CFG is shown in Table 5-316.

Return to the Summary Table.

Table 5-316. APPSS_TPTC_XID_REORDER_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	X	
24	TPTC_B1_DISABLE	R/W	0h	writing 1'b1 will disable the CID-RID-SID reodering feature for TPTC_B1
23-17	RESERVED	R/W	X	
16	TPTC_B0_DISABLE	R/W	0h	writing 1'b1 will disable the CID-RID-SID reodering feature for TPTC_B0
15-9	RESERVED	R/W	X	
8	TPTC_A1_DISABLE	R/W	0h	writing 1'b1 will disable the CID-RID-SID reodering feature for TPTC_A1
7-1	RESERVED	R/W	X	
0	TPTC_A0_DISABLE	R/W	0h	writing 1'b1 will disable the CID-RID-SID reodering feature for TPTC_A0

5.2.3.96 HW_SYNC_FE_CTRL Register (Offset = 33Ch) [Reset = X]

HW_SYNC_FE_CTRL is shown in Table 5-317.

Return to the Summary Table.

Table 5-317. HW_SYNC_FE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	FE2_SEL	R/W	0h	RESERVED
7-1	RESERVED	R/W	X	
0	FE1_SEL	R/W	0h	RESERVED

5.2.3.97 HW_SPARE_REG1 Register (Offset = 340h) [Reset = 00000000h]

HW_SPARE_REG1 is shown in Table 5-318.

Return to the Summary Table.

Table 5-318. HW_SPARE_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Resereved for R&D

5.2.3.98 HW_SPARE_REG2 Register (Offset = 344h) [Reset = 00000000h]

HW_SPARE_REG2 is shown in Table 5-319.

Return to the Summary Table.

Table 5-319. HW_SPARE_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Resereved for R&D

5.2.3.99 HW_SPARE_REG3 Register (Offset = 348h) [Reset = 00000000h]

HW_SPARE_REG3 is shown in Table 5-320.

Return to the Summary Table.

Table 5-320. HW_SPARE_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Resereved for R&D

5.2.3.100 NERROR_MASK Register (Offset = 34Ch) [Reset = X]

NERROR_MASK is shown in Table 5-321.

Return to the Summary Table.

Table 5-321. NERROR_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	MASK	R/W	1h	writing 1'b1 will mask the Nerror propagation to pad Writing 1'b0 will unmask the Nerror propagation to pad

5.2.3.101 HW_SPARE_RW0 Register (Offset = 350h) [Reset = 00000000h]

HW_SPARE_RW0 is shown in Table 5-322.

Return to the Summary Table.

Table 5-322. HW_SPARE_RW0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW0	R/W	0h	Bit 0: Writing 1'b1 will mask the hwa local ram agg serr propagation to ESM Writing 1'b0 will unmask the hwa local ram agg serr propagation to ESM Bit 1 : Writing 1'b1 will mask the hwa local ram agg uerr propagation to ESM Writing 1'b0 will unmask the hwa local ram agg uerr propagation to ESM Bit 2 to 31 Reserved

5.2.3.102 HW_SPARE_RW1 Register (Offset = 354h) [Reset = 00000000h]

HW_SPARE_RW1 is shown in Table 5-323.

Return to the Summary Table.

Table 5-323. HW_SPARE_RW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW1	R/W	0h	Reserved for HW R&D

5.2.3.103 HW_SPARE_RW2 Register (Offset = 358h) [Reset = 00000000h]

HW_SPARE_RW2 is shown in Table 5-324.

Return to the Summary Table.

Table 5-324. HW_SPARE_RW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW2	R/W	0h	Reserved for HW R&D

5.2.3.104 HW_SPARE_RW3 Register (Offset = 35Ch) [Reset = 00000000h]

HW_SPARE_RW3 is shown in Table 5-325.

Return to the Summary Table.

Table 5-325. HW_SPARE_RW3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW3	R/W	0h	Reserved for HW R&D

5.2.3.105 HW_SPARE_RW4 Register (Offset = 360h) [Reset = 00000000h]

HW_SPARE_RW4 is shown in Table 5-326.

Return to the Summary Table.

Table 5-326. HW_SPARE_RW4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW4	R/W	0h	Reserved for HW R&D

5.2.3.106 HW_SPARE_RW5 Register (Offset = 364h) [Reset = 00000000h]

HW_SPARE_RW5 is shown in Table 5-327.

Return to the Summary Table.

Table 5-327. HW_SPARE_RW5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW5	R/W	0h	Reserved for HW R&D

5.2.3.107 HW_SPARE_RO0 Register (Offset = 368h) [Reset = 00000000h]

HW_SPARE_RO0 is shown in Table 5-328.

Return to the Summary Table.

Table 5-328. HW_SPARE_RO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RO0	R	0h	Reserved for HW R&D

5.2.3.108 HW_SPARE_RO1 Register (Offset = 36Ch) [Reset = 00000000h]

HW_SPARE_RO1 is shown in Table 5-329.

Return to the Summary Table.

Table 5-329. HW_SPARE_RO1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RO1	R	0h	Reserved for HW R&D

5.2.3.109 HW_SPARE_RO2 Register (Offset = 370h) [Reset = 00000000h]

HW_SPARE_RO2 is shown in Table 5-330.

Return to the Summary Table.

Table 5-330. HW_SPARE_RO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RO2	R	0h	Reserved for HW R&D

5.2.3.110 HW_SPARE_RO3 Register (Offset = 374h) [Reset = 00000000h]

HW_SPARE_RO3 is shown in Table 5-331.

Return to the Summary Table.

Table 5-331. HW_SPARE_RO3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RO3	R	0h	Reserved for HW R&D

5.2.3.111 HW_SPARE_REC Register (Offset = 378h) [Reset = 00000000h]

HW_SPARE_REC is shown in Table 5-332.

Return to the Summary Table.

Table 5-332. HW_SPARE_REC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	HW_SPARE_REC31	R/W	0h	Reserved for HW R&D
30	HW_SPARE_REC30	R/W	0h	Reserved for HW R&D
29	HW_SPARE_REC29	R/W	0h	Reserved for HW R&D
28	HW_SPARE_REC28	R/W	0h	Reserved for HW R&D

Table 5-332. HW_SPARE_REC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	HW_SPARE_REC27	R/W	0h	Reserved for HW R&D
26	HW_SPARE_REC26	R/W	0h	Reserved for HW R&D
25	HW_SPARE_REC25	R/W	0h	Reserved for HW R&D
24	HW_SPARE_REC24	R/W	0h	Reserved for HW R&D
23	HW_SPARE_REC23	R/W	0h	Reserved for HW R&D
22	HW_SPARE_REC22	R/W	0h	Reserved for HW R&D
21	HW_SPARE_REC21	R/W	0h	Reserved for HW R&D
20	HW_SPARE_REC20	R/W	0h	Reserved for HW R&D
19	HW_SPARE_REC19	R/W	0h	Reserved for HW R&D
18	HW_SPARE_REC18	R/W	0h	Reserved for HW R&D
17	HW_SPARE_REC17	R/W	0h	Reserved for HW R&D
16	HW_SPARE_REC16	R/W	0h	Reserved for HW R&D
15	HW_SPARE_REC15	R/W	0h	Reserved for HW R&D
14	HW_SPARE_REC14	R/W	0h	Reserved for HW R&D
13	HW_SPARE_REC13	R/W	0h	Reserved for HW R&D
12	HW_SPARE_REC12	R/W	0h	Reserved for HW R&D
11	HW_SPARE_REC11	R/W	0h	Reserved for HW R&D
10	HW_SPARE_REC10	R/W	0h	Reserved for HW R&D
9	HW_SPARE_REC9	R/W	0h	Reserved for HW R&D
8	HW_SPARE_REC8	R/W	0h	Reserved for HW R&D
7	HW_SPARE_REC7	R/W	0h	Reserved for HW R&D
6	HW_SPARE_REC6	R/W	0h	Reserved for HW R&D
5	HW_SPARE_REC5	R/W	0h	Reserved for HW R&D
4	HW_SPARE_REC4	R/W	0h	Reserved for HW R&D
3	HW_SPARE_REC3	R/W	0h	Reserved for HW R&D
2	HW_SPARE_REC2	R/W	0h	Reserved for HW R&D
1	HW_SPARE_REC1	R/W	0h	Reserved for HW R&D
0	HW_SPARE_REC0	R/W	0h	Reserved for HW R&D

5.2.3.112 APP_CTRL Register (Offset = 37Ch) [Reset = X]

APP_CTRL is shown in Table 5-333.

Return to the Summary Table.

Table 5-333. APP_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ECC_DISABLE_2K_RAM	R/W	1h	Reserved

5.2.3.113 WIC_CTRL Register (Offset = 380h) [Reset = 00000000h]

WIC_CTRL is shown in Table 5-334.

Return to the Summary Table.

Table 5-334. WIC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WICMASK	R/W	0h	1 => The corresponding interrupt is Masked (interrupt will not be generated) 0 => The corresponding interrupt is UnMasked (interrupt will be generated) 0 : ESM_HI_IRQ (NMI) 1 : ESM_LO_IRQ (INT#1) 2 : FECSS_FRAMETIMER_FRAME_START (INT#33) 3 : MUXED_FECSS_FRAME_START_OFFSET_INTR_TIME1 (INT#35) 4 : FECSS_FRAME_START_OFFSET_INTR_TIME2 (INT#36) 5 : FECSS_FRAME_START_OFFSET_INTR_TIME3 (INT#37) 6 : FECSS_BURST_START_OFFSET_TIME(INT#38) 7 : MUXED_APPSS_RTI1_RTI2_INT_REQ0(INT#43) 8 : MUXED_APPSS_RTI1_RTI2_INT_REQ1(INT#44) 9 : MUXED_APPSS_RTI1_RTI2_INT_REQ2(INT#45) 10 : MUXED_APPSS_RTI1_RTI2_INT_REQ3(INT#46) 11 : APPSS_SPI_IRQ_REQ(INT#14) 12 : SPI2_IRQ_REQ (part of INT#28) 13 : APPSS_LIN_INT0 (INT#10) 14 : APPSS_LIN_INT0 (INT#11) 15 : APPSS_MCAN_INT0(INT#21) 16 : APPSS_MCAN_INT1(INT#22) 17 : APPSS_SCI2_INT0(INT#62) 18 : APPSS_SCI2_INT0(INT#63) 19 : APPSS_SPI_IRQ_REQ(INT#14) 20 : SPI2_IRQ_REQ (part of INT#28) 21 : APPSS_LIN_INT0 (INT#10) 22 : APPSS_LIN_INT0 (INT#11) 23 : APPSS_MCAN_INT0(INT#21) 24 : APPSS_MCAN_INT1(INT#22) 25 : APPSS_SCI2_INT0(INT#62) 26 : APPSS_SCI2_INT0(INT#63) 27 : SYNC_IN 28 : RADAR_DEVICESLEEP_WAKEUP_INTERRUPT 29 to 31 : Reserved

5.2.3.114 WIC_STAT_CLR Register (Offset = 384h) [Reset = 00000000h]

WIC_STAT_CLR is shown in Table 5-335.

Return to the Summary Table.

Table 5-335. WIC_STAT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WICSTATCLR	R/W	0h	1 => Writing 1 to this bit, will clear the WIC_STAT status register of the corresponding bit. Self-clearing 0 => Writing 0 to this bit, leavesWIC_STAT status register unchanged for the corresponding bit.

5.2.3.115 WIC_STAT Register (Offset = 388h) [Reset = 00000000h]

WIC_STAT is shown in Table 5-336.

Return to the Summary Table.

Table 5-336. WIC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WICSTAT	R	0h	1 => Interrupt bit set. The interupt bit is sticky bit. Should be cleared using WIC_STAT_CLR register or subsystem reset. 0 -> Interrupt bit not set. Sticky bits keep their value when they changed to logical 1 and is cleared only by writing 1 to WIC_STAT_CLR register.

5.2.3.116 WICEN Register (Offset = 38Ch) [Reset = X]

WICEN is shown in Table 5-337.

Return to the Summary Table.

Table 5-337. WICEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	WICEN	R/W	1h	1 :> Wakeup Interrupt Controller (WIC) of CM4 is Enabled 0 :> Disabled

5.2.3.117 FORCEFCLKACTIVE Register (Offset = 390h) [Reset = X]

FORCEFCLKACTIVE is shown in Table 5-338.

Return to the Summary Table.

Table 5-338. FORCEFCLKACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	FORCEFCLKACTIVE	R/W	0h	1 :> Forces FCLK to be active and inhibits CM4 Entering CPU DeepSleep mode 0 :> Allows gating of FCLK based on CPU DEEPSLEEP entry mechanism

5.2.3.118 FECSS_CLK_GATE Register (Offset = 394h) [Reset = X]

FECSS_CLK_GATE is shown in Table 5-339.

Return to the Summary Table.

Table 5-339. FECSS_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	X	
5-3	GRP2	R/W	0h	Multibit: Writing 3'b111 will gate ADC_CLK going to DFE and Timing Engine
2-0	GRP1	R/W	0h	Multibit: Writing 3'b111 will gate FEC_SYS_CLK and FECSS peripheral clocks except DFE and Timing Engine

5.2.3.119 APPSS_SHARED_MEM_CLK_GATE Register (Offset = 398h) [Reset = X]

APPSS_SHARED_MEM_CLK_GATE is shown in Table 5-340.

Return to the Summary Table.

Table 5-340. APPSS_SHARED_MEM_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	MEM1_APP_ENABLE	R/W	1h	1'b 1 : Enable APP CLK ICG for second 128KB of shared mem 1'b 0 : Disable APP CLK ICG for second 128 KB of shared mem
2	MEM1_HWA_ENABLE	R/W	1h	1'b 1 : Enable HWA CLK ICG for second 128 KB of shared mem 1'b 0 : Disable HWA CLK ICG for second 128 KB of shared mem
1	MEM0_APP_ENABLE	R/W	1h	1'b 1 : Enable APP CLK ICG for first 128KB of shared mem 1'b 0 : Disable APP CLK ICG for first 128 KB of shared mem
0	MEM0_HWA_ENABLE	R/W	1h	1'b 1 : Enable HWA CLK ICG for first 128 KB of shared mem 1'b 0 : Disable HWA CLK ICG for first 128 KB of shared mem

5.2.3.120 APPSS_MEM_INIT_SLICE_SEL Register (Offset = 39Ch) [Reset = X]

APPSS_MEM_INIT_SLICE_SEL is shown in Table 5-341.

Return to the Summary Table.

Table 5-341. APPSS_MEM_INIT_SLICE_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4-3	CFG_BANK2	R/W	3h	Selects the APPSS RAM2 partition that needs to be initialized. More than 1 partitions can be selected for mem_init. Bit# 0 : Selects RAM_2A (16KB) Bit# 1 : Selects RAM_2B (112KB) 1 => RAM partition selected for mem_init operation 0 => RAM partition not selected for mem_init operation.
2-0	CFG_BANK1	R/W	7h	Selects the APPSS RAM1 partition that needs to be initialized. More than 1 partitions can be selected for mem_init. Bit# 0 : Selects RAM_1A (64KB) Bit# 1 : Selects RAM_1B (64KB) Bit# 2 : Selects RAM_1C (128KB) 1 => RAM selected for mem_init operation 0 => RAM not selected for mem_init operation.

5.2.3.121 APPSS_QSPI_CHAR_EXT_CLK_EN Register (Offset = 3A0h) [Reset = X]

APPSS_QSPI_CHAR_EXT_CLK_EN is shown in Table 5-342.

Return to the Summary Table.

Table 5-342. APPSS_QSPI_CHAR_EXT_CLK_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ENABLE	R/W	0h	Selects the QSPI system clock. Only for DFT purposes. This should not be changed for functional operation. 0 => QSPI_CLK from APPSS RCM 1 => SPI1_CLK from APPSS RCM

5.2.3.122 APPSS_QSPI_EXT_CLK_EN Register (Offset = 3A4h) [Reset = X]

APPSS_QSPI_EXT_CLK_EN is shown in Table 5-343.

Return to the Summary Table.

Table 5-343. APPSS_QSPI_EXT_CLK_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ENABLE	R/W	0h	Selects the QSPI interface clock. This register bit is used only for AC CHAR operation and not for functional usage. 0 => default QSPI IP clock return from PAD 1 => SPI1 IF CLK. (McSPI IF clock).

5.2.3.123 SPI1_SMART_IDLE Register (Offset = 3A8h) [Reset = X]

SPI1_SMART_IDLE is shown in Table 5-344.

Return to the Summary Table.

Table 5-344. SPI1_SMART_IDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	X	
5	WAKEUP_RAW	R	0h	Description: RAW status of CLKSTOP_WAKEUP from SPI1 module. This should be interpreted along with SPI1_SMART_IDLE_WAKEUP SPI1_SMART_IDLE_WAKEUP_RAW, SPI1_SMART_IDLE_WAKEUP 0 , 0 => WAKEUP is LOW from IP, and No pending WAKEUP status 0 , 1 => WAKEUP is LOW from IP, and pending WAKEUP status 1 , 0 => WAKEUP is HIGH from IP, and No pending WAKEUP status 1 , 1 => WAKEUP is HIGH from IP, and pending WAKEUP status
4	ACK_RAW	R	0h	Description: RAW status of CLKSTOP_ACK from McSPI (SPI1) module. This should be interpreted along with SPI1_SMART_IDLE_ACK SPI1_SMART_IDLE_ACK_RAW, SPI1_SMART_IDLE_ACK 0 , 0 => ACK is LOW from IP, and No pending ACK status 0 , 1 => ACK is LOW from IP, and pending ACK status 1 , 0 => ACK is HIGH from IP, and No pending ACK status 1 , 1 => ACK is HIGH from IP, and pending ACK status
3	WAKEUP	R/W	0h	This register reflects the Wakeup Status of the IP. The bit is sticky bit and the user is should clear once the status is read by write- 1-to-clear.
2	AUTO_EN	R/W	0h	It is used to select smart idle mode. 1 => Automatic mode - Entry to smart idle mode Is manual by setting SMART_IDLE_ENABLE = 1. When the wakeup Signal is asserted (based on the activity), The clkstop_req is pulled low automatically. 0 => Manual mode - The entry and exit to Smart Idle is user controlled based on polling SMART_IDLE_ACK and SMART_IDLE_WAKEUP
1	ACK	R/W	0h	1 => SPI1 in smart idle mode 0 => SPI1 not in smart idle mode The bit is sticky bit and the user is should clear once the status is read by write- 1-to-clear.
0	ENABLE	R/W	0h	1 => Smart IDLE mode enabled. When set, request the clock gating of SPI1 module. 0 => Disable Smart IDLE mode for SPI1

5.2.3.124 SPI2_SMART_IDLE Register (Offset = 3ACh) [Reset = X]

SPI2_SMART_IDLE is shown in Table 5-345.

Return to the Summary Table.

Table 5-345. SPI2_SMART_IDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	X	
5	WAKEUP_RAW	R	0h	Description: RAW status of CLKSTOP_WAKEUP from SPI2 module. This should be interpreted along with SPI2_SMART_IDLE_WAKEUP SPI2_SMART_IDLE_WAKEUP_RAW, SPI2_SMART_IDLE_WAKEUP 0 , 0 => WAKEUP is LOW from IP, and No pending WAKEUP status 0 , 1 => WAKEUP is LOW from IP, and pending WAKEUP status 1 , 0 => WAKEUP is HIGH from IP, and No pending WAKEUP status 1 , 1 => WAKEUP is HIGH from IP, and pending WAKEUP status
4	ACK_RAW	R	0h	Description: RAW status of CLKSTOP_ACK from McSPI (SPI2) module. This should be interpreted along with SPI2_SMART_IDLE_ACK SPI2_SMART_IDLE_ACK_RAW, SPI2_SMART_IDLE_ACK 0 , 0 => ACK is LOW from IP, and No pending ACK status 0 , 1 => ACK is LOW from IP, and pending ACK status 1 , 0 => ACK is HIGH from IP, and No pending ACK status 1 , 1 => ACK is HIGH from IP, and pending ACK status

Table 5-345. SPI2_SMART_IDLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	WAKEUP	R/W	0h	This register reflects the Wakeup Status of the IP. The bit is sticky bit and the user is should clear once the status is read by write- 1-to-clear.
2	AUTO_EN	R/W	0h	It is used to select smart idle mode. 1 => Automatic mode - In this mode, entry to smart idle mode is manual by setting SMART_IDLE_ENABLE = 1. When the wakeup Signal is asserted (based on the activity), The clkstop_req is pulled low automatically. 0 => Manual mode - The entry and exit to Smart Idle is user controlled based on polling SMART_IDLE_ACK and SMART_IDLE_WAKEUP
1	ACK	R/W	0h	1 => SPI2 in smart idle mode 0 => SPI2 not in smart idle mode The bit is sticky bit and the user is should clear once the status is read by write- 1-to-clear.
0	ENABLE	R/W	0h	1 => Smart IDLE mode enabled. When set, request the clock gating of SPI2 module. 0 => Disable Smart IDLE mode for SPI2

5.2.3.125 CAN_SMART_IDLE Register (Offset = 3B0h) [Reset = X]

CAN_SMART_IDLE is shown in Table 5-346.

Return to the Summary Table.

Table 5-346. CAN_SMART_IDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	X	
5	WAKEUP_RAW	R	0h	Description: RAW status of CLKSTOP_WAKEUP from CANFD module. This should be interpreted along with CAN_SMART_IDLE_WAKEUP CAN_SMART_IDLE_WAKEUP_RAW, CAN_SMART_IDLE_WAKEUP 0 , 0 => WAKEUP is LOW from IP, and No pending WAKEUP status 0 , 1 => WAKEUP is LOW from IP, and pending WAKEUP status 1 , 0 => WAKEUP is HIGH from IP, and No pending WAKEUP status 1 , 1 => WAKEUP is HIGH from IP, and pending WAKEUP status
4	ACK_RAW	R	0h	Description: RAW status of CLKSTOP_ACK from CANFD module. This should be interpreted along with CAN_SMART_IDLE_ACK CAN_SMART_IDLE_ACK_RAW, CAN_SMART_IDLE_ACK 0 , 0 => ACK is LOW from IP, and No pending ACK status 0 , 1 => ACK is LOW from IP, and pending ACK status 1 , 0 => ACK is HIGH from IP, and No pending ACK status 1 , 1 => ACK is HIGH from IP, and pending ACK status
3	WAKEUP	R/W	0h	This register reflects the Wakeup Status of the IP. The bit is sticky bit and the user is should clear once the status is read by write- 1-to-clear.
2	AUTO_EN	R/W	0h	It is used to select smart idle mode. 1 => Automatic mode - In this mode, entry to smart idle mode is manual by setting SMART_IDLE_ENABLE = 1. When the wakeup Signal is asserted (based on the activity), The clkstop_req is pulled low automatically. 0 => Manual mode - The entry and exit to Smart Idle is user controlled based on polling SMART_IDLE_ACK and SMART_IDLE_WAKEUP
1	ACK	R/W	0h	1 => CAN in smart idle mode 0 => CAN not in smart idle mode The bit is sticky bit and the user is should clear once the status is read by write- 1-to-clear.

Table 5-346. CAN_SMART_IDLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ENABLE	R/W	0h	1 => Smart IDLE mode enabled. When set, Request the clock gating of CAN module. 0 => Disable Smart IDLE mode for CAN

5.2.3.126 LIN_SMART_IDLE Register (Offset = 3B4h) [Reset = X]

LIN_SMART_IDLE is shown in Table 5-347.

Return to the Summary Table.

Table 5-347. LIN_SMART_IDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	ACK	R	0h	1 => LIN in smart idle mode 0 => LIN not in smart idle mode
0	ENABLE	R/W	0h	1 => Smart IDLE mode enabled. When set, Request the clock gating of LIN module. 0 => Disable Smart IDLE mode for LIN

5.2.3.127 HWASS_CLK_GATE Register (Offset = 3B8h) [Reset = X]

HWASS_CLK_GATE is shown in Table 5-348.

Return to the Summary Table.

Table 5-348. HWASS_CLK_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	ENABLE	R/W	0h	RESERVED

5.2.3.128 CFG_TIMEOUT_PCR3 Register (Offset = 3BCh) [Reset = 00000FFFh]

CFG_TIMEOUT_PCR3 is shown in Table 5-349.

Return to the Summary Table.

Table 5-349. CFG_TIMEOUT_PCR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VALUE	R/W	FFFh	PCR3Timeout Value

5.2.3.129 RESERVED0 Register (Offset = 3C0h) [Reset = X]

RESERVED0 is shown in Table 5-350.

Return to the Summary Table.

Table 5-350. RESERVED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	X	
15-0	RESERVED	R/W	0h	RESERVED - POR Reset

5.2.3.130 APPSS_ERRAGG_MASK1 Register (Offset = 3C4h) [Reset = X]

APPSS_ERRAGG_MASK1 is shown in Table 5-351.

Return to the Summary Table.

Table 5-351. APPSS_ERRAGG_MASK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	X	
11	CLUSTER12_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster12_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
10	CLUSTER11_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster11_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
9	CLUSTER10_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster10_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
8	CLUSTER9_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster9_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7	CLUSTER8_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster8_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	CLUSTER7_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster7_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	CLUSTER6_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster6_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	CLUSTER5_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster5_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
3	CLUSTER4_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster4_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	CLUSTER3_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster3_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	CLUSTER2_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster2_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	CLUSTER1_POWER_DOWN_ACCESS_ERR	R/W	0h	Mask Interrupt from cluster1_power_down_access_err Interrupt APPSS_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

5.2.3.131 APPSS_ERRAGG_STATUS1 Register (Offset = 3C8h) [Reset = X]

APPSS_ERRAGG_STATUS1 is shown in Table 5-352.

Return to the Summary Table.

Table 5-352. APPSS_ERRAGG_STATUS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	X	
11	CLUSTER12_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster12_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
10	CLUSTER11_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster11_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
9	CLUSTER10_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster10_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
8	CLUSTER9_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster9_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
7	CLUSTER8_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster8_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
6	CLUSTER7_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster7_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
5	CLUSTER6_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster6_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
4	CLUSTER5_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster5_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
3	CLUSTER4_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster4_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
2	CLUSTER3_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster3_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
1	CLUSTER2_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster2_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.
0	CLUSTER1_POWER_DOWN_ACCESS_ERR	R/W	0h	Status of Interrupt from cluster1_power_down_access_err Set only if Interupt is unmasked in APPSS_ERRAGG_MASK1 Wrie 0x1 to clear this interrupt.

5.2.3.132 FORCEHCLKACTIVE Register (Offset = 3CCh) [Reset = X]

FORCEHCLKACTIVE is shown in Table 5-353.

Return to the Summary Table.

Table 5-353. FORCEHCLKACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	FORCEHCLKACTIVE	R/W	0h	1 :> Gate HCLK 0 :> UnGate HCLK

5.2.3.133 APPSS_RAM1_OWRITE_ERR Register (Offset = 3D0h) [Reset = X]

APPSS_RAM1_OWRITE_ERR is shown in Table 5-354.

Return to the Summary Table.

Table 5-354. APPSS_RAM1_OWRITE_ERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ERR	R/W	0h	RAM1 ahb2sram write error - Sticky Bit

5.2.3.134 APPSS_RAM1_OWRITE_ERR_ADDR Register (Offset = 3D4h) [Reset = 00000000h]

APPSS_RAM1_OWRITE_ERR_ADDR is shown in Table 5-355.

Return to the Summary Table.

Table 5-355. APPSS_RAM1_OWRITE_ERR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	RAM1 ahb2sram write error Address

5.2.3.135 APPSS_RAM2_OWRITE_ERR Register (Offset = 3D8h) [Reset = X]

APPSS_RAM2_OWRITE_ERR is shown in Table 5-356.

Return to the Summary Table.

Table 5-356. APPSS_RAM2_OWRITE_ERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ERR	R/W	0h	RAM2 ahb2sram write error - Sticky Bit

5.2.3.136 APPSS_RAM2_OWRITE_ERR_ADDR Register (Offset = 3DCh) [Reset = 00000000h]

APPSS_RAM2_OWRITE_ERR_ADDR is shown in Table 5-357.

Return to the Summary Table.

Table 5-357. APPSS_RAM2_OWRITE_ERR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	RAM2 ahb2sram write error Address

5.2.3.137 APPSS_RAM3_OWRITE_ERR Register (Offset = 3E0h) [Reset = X]

APPSS_RAM3_OWRITE_ERR is shown in Table 5-358.

Return to the Summary Table.

Table 5-358. APPSS_RAM3_OWRITE_ERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ERR	R/W	0h	RAM3 ahb2sram write error - Sticky Bit

5.2.3.138 APPSS_RAM3_OWRITE_ERR_ADDR Register (Offset = 3E4h) [Reset = 00000000h]

APPSS_RAM3_OWRITE_ERR_ADDR is shown in Table 5-359.

Return to the Summary Table.

Table 5-359. APPSS_RAM3_OWRITE_ERR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	RAM3 ahb2sram write error Address

5.2.3.139 APPSS_SHRD_RAM_OWRITE_ERR Register (Offset = 3E8h) [Reset = X]

APPSS_SHRD_RAM_OWRITE_ERR is shown in Table 5-360.

Return to the Summary Table.

Table 5-360. APPSS_SHRD_RAM_OWRITE_ERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ERR	R/W	0h	SHARED RAM ahb2sram write error - Sticky Bit

5.2.3.140 APPSS_SHRD_RAM_OWRITE_ERR_ADDR Register (Offset = 3ECh) [Reset = 00000000h]

APPSS_SHRD_RAM_OWRITE_ERR_ADDR is shown in Table 5-361.

Return to the Summary Table.

Table 5-361. APPSS_SHRD_RAM_OWRITE_ERR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	SHARED RAM ahb2sram write error Address

5.2.3.141 APPSS_OWRITE_ERR_AGGR Register (Offset = 3F0h) [Reset = X]

APPSS_OWRITE_ERR_AGGR is shown in Table 5-362.

Return to the Summary Table.

Table 5-362. APPSS_OWRITE_ERR_AGGR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	ERR	R/W	0h	Ored error of all write error signals -Sticky Bit

5.2.3.142 HW_SPARE_RW6 Register (Offset = 3F4h) [Reset = 00000000h]

HW_SPARE_RW6 is shown in Table 5-363.

Return to the Summary Table.

Table 5-363. HW_SPARE_RW6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW6	R/W	0h	Reserved for HW R&D

5.2.3.143 HW_SPARE_RW7 Register (Offset = 3F8h) [Reset = 00000000h]

HW_SPARE_RW7 is shown in Table 5-364.

Return to the Summary Table.

Table 5-364. HW_SPARE_RW7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW7	R/W	0h	Reserved for HW R&D

5.2.3.144 HW_SPARE_RW8 Register (Offset = 3FCh) [Reset = 00000000h]

HW_SPARE_RW8 is shown in Table 5-365.

Return to the Summary Table.

Table 5-365. HW_SPARE_RW8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW8	R/W	0h	Reserved for HW R&D

5.2.3.145 HW_SPARE_RW9 Register (Offset = 400h) [Reset = 00000000h]

HW_SPARE_RW9 is shown in Table 5-366.

Return to the Summary Table.

Table 5-366. HW_SPARE_RW9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_RW9	R/W	0h	Reserved for HW R&D

5.2.3.146 HW_SPARE_HWA_RW0 Register (Offset = 404h) [Reset = 00000000h]

HW_SPARE_HWA_RW0 is shown in Table 5-367.

Return to the Summary Table.

Table 5-367. HW_SPARE_HWA_RW0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HW_SPARE_HWA_RW0	R/W	0h	Reserved for HW R&D

5.2.3.147 LOCK0_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0_KICK0 is shown in Table 5-368.

Return to the Summary Table.

- KICK0 component

Table 5-368. LOCK0_KICK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_KICK0	R/W	0h	- KICK0 component

5.2.3.148 LOCK0_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0_KICK1 is shown in Table 5-369.

Return to the Summary Table.

- KICK1 component

Table 5-369. LOCK0_KICK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_KICK1	R/W	0h	- KICK1 component

5.2.3.149 INTR_RAW_STATUS Register (Offset = 1010h) [Reset = X]

INTR_RAW_STATUS is shown in Table 5-370.

Return to the Summary Table.

Interrupt Raw Status/Set Register

Table 5-370. INTR_RAW_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	PROXY_ERR	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	KICK_ERR	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	ADDR_ERR	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	PROT_ERR	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

ADVANCE INFORMATION

5.2.3.150 INTR_ENABLED_STATUS_CLEAR Register (Offset = 1014h) [Reset = X]

INTR_ENABLED_STATUS_CLEAR is shown in Table 5-371.

Return to the Summary Table.

Interrupt Enabled Status/Clear register

Table 5-371. INTR_ENABLED_STATUS_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	ENABLED_PROXY_ERR	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	ENABLED_KICK_ERR	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	ENABLED_ADDR_ERR	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	ENABLED_PROT_ERR	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

5.2.3.151 INTR_ENABLE Register (Offset = 1018h) [Reset = X]

INTR_ENABLE is shown in Table 5-372.

Return to the Summary Table.

Interrupt Enable register

Table 5-372. INTR_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	

Table 5-372. INTR_ENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PROXY_ERR_EN	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	KICK_ERR_EN	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	ADDR_ERR_EN	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	PROT_ERR_EN	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

5.2.3.152 INTR_ENABLE_CLEAR Register (Offset = 101Ch) [Reset = X]

INTR_ENABLE_CLEAR is shown in Table 5-373.

Return to the Summary Table.

Interrupt Enable Clear register

Table 5-373. INTR_ENABLE_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	PROXY_ERR_EN_CLR	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	KICK_ERR_EN_CLR	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	ADDR_ERR_EN_CLR	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	PROT_ERR_EN_CLR	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

5.2.3.153 EOI Register (Offset = 1020h) [Reset = X]

EOI is shown in Table 5-374.

Return to the Summary Table.

EOI register

Table 5-374. EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	EOI_VECTOR	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

5.2.3.154 FAULT_ADDRESS Register (Offset = 1024h) [Reset = 00000000h]

FAULT_ADDRESS is shown in Table 5-375.

Return to the Summary Table.

Fault Address register

Table 5-375. FAULT_ADDRESS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FAULT_ADDR	R	0h	Fault Address.

5.2.3.155 FAULT_TYPE_STATUS Register (Offset = 1028h) [Reset = X]

FAULT_TYPE_STATUS is shown in Table 5-376.

Return to the Summary Table.

Fault Type Status register

Table 5-376. FAULT_TYPE_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	X	
6	FAULT_NS	R	0h	Non-secure access.
5-0	FAULT_TYPE	R	0h	Fault Type 10_ 0000 = Supervisor read fault - priv = 1 dir = 1 dtype != 1 01_ 0000 = Supervisor write fault - priv = 1 dir = 0 00_ 1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_ 0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_ 0010 = User write fault - priv = 0 dir = 0 00_ 0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_ 0000 = No fault

5.2.3.156 FAULT_ATTR_STATUS Register (Offset = 102Ch) [Reset = 00000000h]

FAULT_ATTR_STATUS is shown in Table 5-377.

Return to the Summary Table.

Fault Attribute Status register

Table 5-377. FAULT_ATTR_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	FAULT_XID	R	0h	XID.
19-8	FAULT_ROUTEID	R	0h	Route ID.
7-0	FAULT_PRIVID	R	0h	Privilege ID.

5.2.3.157 FAULT_CLEAR Register (Offset = 1030h) [Reset = X]

FAULT_CLEAR is shown in Table 5-378.

Return to the Summary Table.

Fault Clear register

Table 5-378. FAULT_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	X	
0	FAULT_CLR	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

5.3 Device Clock Architecture

5.3.1 Clock Overview

Figure 5-2 shows a high-level overview of the device clock architecture. Figure 5-2 captures the key clock sources and the configuration options available to select the appropriate clock source.

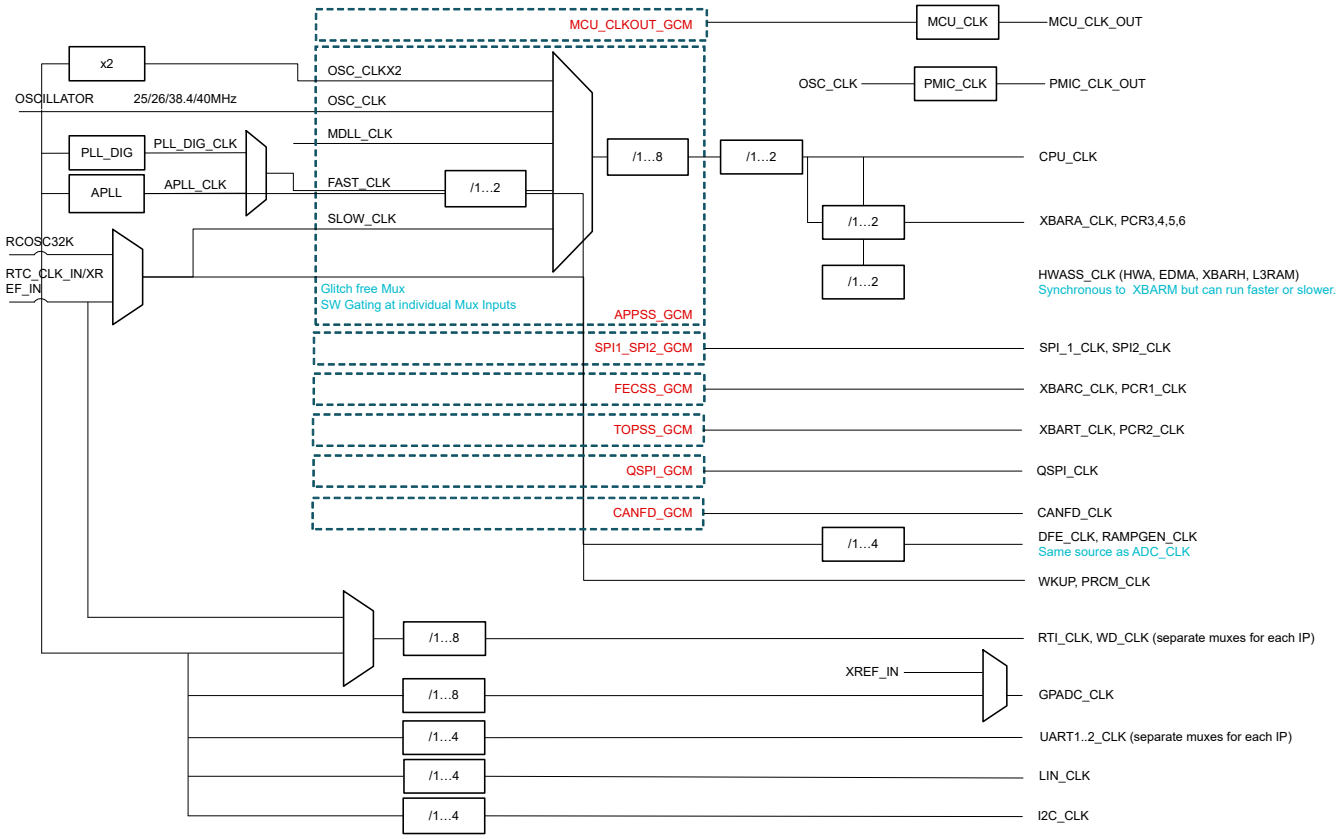


Figure 5-2. Clock Tree Configuration

5.3.2 Clock Selection

Table 5-379 lists the configuration options for the clock source, divider, and gating selections for different peripheral clocks.

ADVANCE INFORMATION

Table 5-379. Clock Selection

Clock Mux	Clock Sources		CLKSRCSEL Control	CLKSRCSEL Control	CLKDIV Control	CLKGATE Control
APP_CPU_CLKCTL_SRCSEL	0	OSC_CLK	APP_CPU_CLKCTL Register Field DescriptionsAPP_CPU_CLKCTL_SELECTED	APP_CPU_CLKCTL Register Field DescriptionsAPP_CPU_CLKCTL_SRCSEL	APP_CPU_CLKCTL Register Field DescriptionsAPP_CPU_CLKCTL_DIVR	NA
	1	SLOW_CLK				
	2	MDLL_CLK				
	3	FAST_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				
TOPSS_CLKCTL_SRCSEL (XBART, PCR2)	0	OSC_CLK	TOPSS_CLKCTL Register Field DescriptionsTOPSS_CLKCTL_SELECTED	TOPSS_CLKCTL Register Field DescriptionsTOPSS_CLKCTL_SRCSEL	TOPSS_CLKCTL Register Field DescriptionsTOPSS_CLKCTL_DIVR	TOPSS_CLKCTL Register Field DescriptionsTOPSS_CLKCTL_GATE
	1	SLOW_CLK				
	2	MDLL_CLK				
	3	FAST_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				

ADVANCE INFORMATION

Table 5-379. Clock Selection (continued)

Clock Mux	Clock Sources		CLKSRCSEL Control	CLKSRCSEL Control	CLKDIV Control	CLKGATE Control
APP_SPI_CLKCTL_SRCSEL	0	OSC_CLK	APP_SPI_CLKCTL Register Field DescriptionsAPP_SPI_CLKCTL_SELECTED	APP_SPI_CLKCTL Register Field DescriptionsAPP_SPI_CLKCTL_SRCSEL	APP_SPI_CLKCTL Register Field DescriptionsAPP_SPI_CLKCTL_DIVR	APP_SPI_CLKCTL Register Field DescriptionsAPP_SPI_CLKCTL_GATE
	1	OSC_CLKX2				
	2	MDLL_CLK				
	3	FAST_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				
APP_CAN_CLKCTL_SRCSEL	0	OSC_CLK	APP_CAN_CLKCTL Register Field DescriptionsAPP_CAN_CLKCTL_SELECTED	APP_CAN_CLKCTL Register Field DescriptionsAPP_CAN_CLKCTL_SRCSEL	APP_CAN_CLKCTL Register Field DescriptionsAPP_CAN_CLKCTL_DIVR	APP_CAN_CLKCTL Register Field DescriptionsAPP_CAN_CLKCTL_GATE
	1	OSC_CLKX2				
	2	MDLL_CLK				
	3	FAST_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				

Table 5-379. Clock Selection (continued)

Clock Mux	Clock Sources		CLKSRCSEL Control	CLKSRCSEL Control	CLKDIV Control	CLKGATE Control
APP_QSPI_CLKCTL_SRCSEL	0	OSC_CLK	APP_QSPI_CLKCTL Register Field DescriptionsAPP_QSPI_CLKCTL_SELECTED	APP_QSPI_CLKCTL Register Field DescriptionsAPP_QSPI_CLKCTL_SRCSEL	APP_QSPI_CLKCTL Register Field DescriptionsAPP_QSPI_CLKCTL_DIVR	APP_QSPI_CLKCTL Register Field DescriptionsAPP_QSPI_CLKCTL_GATE
	1	OSC_CLKX2				
	2	MDLL_CLK				
	3	FAST_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				
APP_RTI_CLKCTL_SRCSEL	0	OSC_CLK	APP_RTI_CLKCTL Register Field DescriptionsAPP_RTI_CLKCTL_SELECTED	APP_RTI_CLKCTL Register Field DescriptionsAPP_RTI_CLKCTL_SRCSEL	APP_RTI_CLKCTL Register Field DescriptionsAPP_RTI_CLKCTL_DIVR	APP_RTI_CLKCTL Register Field DescriptionsAPP_RTI_CLKCTL_GATE
	1	XREF_IN				
	2	OSC_CLK				
	3	SLOW_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				

ADVANCE INFORMATION

Table 5-379. Clock Selection (continued)

Clock Mux	Clock Sources		CLKSRCSEL Control	CLKSRCSEL Control	CLKDIV Control	CLKGATE Control
APP_WD_CLKCTL_SRCSEL	0	OSC_CLK	APP_WD_CLKCTL Register Field DescriptionsAPP_WD_CLKCTL_SELECTED	APP_WD_CLKCTL Register Field DescriptionsAPP_WD_CLKCTL_SRCSEL	APP_WD_CLKCTL Register Field DescriptionsAPP_WD_CLKCTL_DIVR	APP_WD_CLKCTL Register Field DescriptionsAPP_WD_CLKCTL_GATE
	1	XREF_IN				
	2	OSC_CLK				
	3	SLOW_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				
MCU_CLKOUT_GCM_CLKSRC_SEL	0	OSC_CLK	MCUCLKOUT_CLKCTL Register Field DescriptionsMCU_CLKOUT_SELECTED	MCUCLKOUT_CLKCTL Register Field DescriptionsMCUCLKOUT_CLKCTL_MCUCLKOUT_CLK_SRC_SEL	MCUCLKOUT_CLKCTL Register Field DescriptionsMCUCLKOUT_CLKCTL_MCUCLKOUT_CLK_DIVR	Table 5-134MCUCLKOUT_CLKCTL_MCUCLKOUT_CLK_SW_GATE
	1	MDLL_CLK				
	2	FAST_CLK				
	3	SLOW_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				
SLOW_CLK_SRCSEL	0	RCOSC32K	SLOW_CLK_CLKCTL Register Field DescriptionsSLOW_CLK	SLOW_CLK_CLKCTL Register Field DescriptionsSLOW_CLK_CLKCTL_SLOW_CLK_SRC_SEL	NA	NA
	1	RTC_CLK_IN				
	2					
	3					
	4					
	5					
	6					
	7					

Table 5-379. Clock Selection (continued)

Clock Mux	Clock Sources		CLKSRCSEL Control	CLKSRCSEL Control	CLKDIV Control	CLKGATE Control
DEBUGSS_CLK_SRCSEL	0	SLOW_CLK	DEBUGSS_CLK_CLKCTL Register Field DescriptionsDEBUGSS_CLK_SELECTED	DEBUGSS_CLK_CLKCTL Register Field DescriptionsDEBUGSS_CLK_CLKCTL_DEBUGSS_CLK_SRC_SEL	NA	NA
	1	RCOSC10M				
	2	TOPSS_CLK				
	3	SLOW_CLK				
	4	SLOW_CLK				
	5	SLOW_CLK				
	6	SLOW_CLK				
	7	SLOW_CLK				

5.3.3 APLL Clock Frequencies

FREF MHz	Mult Fact	VCO FREQ MHz	APLL CLK to DIG MHz VCO/10 MHz	System Clock Digital
25	320	8000	800	160
26	310	8060	806	161.2
38.4	210	8064	806.4	161.28
40	200	8000	800	160

5.3.4 PLL_DIG Clock Frequency and Configuration Sequence

The DIG PLL can give out close to 320 MHz CLK for the CPU based on the divider settings and the XTAL freq. Table 5-380 gives a snapshot for different frequencies.

Table 5-380. PLL_DIG Clock Frequency

XTAL	10M	25M	26M	38.4M	40M
NDIV	5	10	20	25	20
MDIV	160	128	248	210	160
Fout= XTAL*MDIV/NDIV	320M	320M	320M	322.56M	320M
Fout/2	160M	160M	161.2M	161.28M	160M

If you are using the APLL CLK for the CPU CLK, Table 5-381 lists the output frequencies.

Table 5-381. APLL CLK Clock Frequency

XTAL	10M	25M	26M	38.4M	40M
M	800	320	308	210	200
APLL CLK= XTAL*M	8000M	8000M	8060M	8064	8000M
Fout=APLLCLK/50	160M	160M	161.2M	161.28	160M

5.3.5 APLL_CTRL Registers

Table 5-382 lists the memory-mapped registers for the APLL_CTRL registers. All register offset addresses not listed in Table 5-382 should be considered as reserved locations and the register contents should not be modified.

Table 5-382. APLL_CTRL Registers

Offset	Acronym	Register Name	Section
0h	PID	PID register	Go
4h	CLK_APLL_STATUS_REG1		Go
8h	CLK_APLL_STATUS_REG2		Go
Ch	CLK_CTRL_REG1_APLL		Go
10h	CLK_CTRL_REG10_APLL		Go
14h	CLK_CTRL_REG11_APLL		Go
18h	CLK_CTRL_REG2_APLL		Go
1Ch	CLK_CTRL_REG2_LDO_CLKTOP		Go
20h	CLK_CTRL_REG3_APLL		Go
24h	CLK_CTRL_REG3_LDO_CLKTOP		Go
28h	CLK_CTRL_REG4_APLL		Go
2Ch	CLK_CTRL_REG4_LDO_CLKTOP		Go
30h	CLK_CTRL_REG5_APLL		Go
34h	CLK_CTRL_REG6_APLL		Go
38h	CLK_CTRL_REG7_APLL		Go
3Ch	CLK_CTRL_REG8_APLL		Go
40h	CLK_CTRL_REG9_APLL		Go
44h	CLK_MDLL_REG1		Go
48h	CLK_STATUS_REG		Go
4Ch	CLK_XTAL_X2_REG1		Go
50h	REFSYS_CTRL_REG0_LOWV		Go
54h	WU_SPARE_OUT_LOWV		Go
58h	WU_STATUS_REG_LOWV		Go
5Ch	ANALOG_WU_STATUS_REG_POLARITY_INV		Go
60h	ANALOG_CLK_STATUS_REG_POLARITY_INV		Go
64h	ANALOG_WU_STATUS_REG_MASK		Go
68h	ANALOG_CLK_STATUS_REG_MASK		Go
6Ch	ANALOG_CLK_GOOD_STATUS		Go
70h	ANALOG_CLK_GOOD_MASK		Go
1008h	LOCK0_KICK0	- KICK0 component	Go
100Ch	LOCK0_KICK1	- KICK1 component	Go
1010h	intr_raw_status	Interrupt Raw Status/Set Register	Go
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	Go
1018h	intr_enable	Interrupt Enable register	Go
101Ch	intr_enable_clear	Interrupt Enable Clear register	Go
1020h	eoi	EOI register	Go
1024h	fault_address	Fault Address register	Go
1028h	fault_type_status	Fault Type Status register	Go
102Ch	fault_attr_status	Fault Attribute Status register	Go
1030h	fault_clear	Fault Clear register	Go

Complex bit access types are encoded to fit into small table cells. Table 5-383 shows the codes that are used for access types in this section.

Table 5-383. APLL_CTRL Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

5.3.5.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in Table 5-384.

Return to the Summary Table.

PID register

Table 5-384. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

5.3.5.2 CLK_APLL_STATUS_REG1 Register (Offset = 4h) [Reset = 00000000h]

CLK_APLL_STATUS_REG1 is shown in Table 5-385.

Return to the Summary Table.

Table 5-385. CLK_APLL_STATUS_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	SPARE4	R	0h	Spare bit
28-24	VCO_MID_SELECTED_BANK_VAL	R	0h	Mid code value (binary value for all banks) for the selected bank
23-21	SPARE3	R	0h	Spare bit
20-16	VCO_PVT_BANK_CODE_FINAL	R	0h	Final calculated Coarse (PVT) Bank Code value being applied to Coarse (PVT) Bank - Binary for Coarse (PVT)
15	SPARE2	R	0h	Spare bit
14-12	VCO_CAP_BANK_FSM_STATE	R	0h	BANK FSM State value for Debug
11	SPARE1	R	0h	Spare bit
10-8	ITER_FSM_STATE	R	0h	ITERATION FSM State value for Debug

Table 5-385. CLK_APLL_STATUS_REG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	VCO_CAP_BANK_SELECTED	R	0h	Current Selected Bank: 00: None, 01: PVT, 11: AB, 10: FINE
5-4	FCW_CALC_FSM_STATE	R	0h	FCW Calculation FSM State value for Debug
3	SPARE0	R	0h	Spare bit
2	ITER_FSM_DONE	R	0h	Iteration FSM done strobe indication
1	VCO_SETTLING_COUNT_DONE	R	0h	VCO Settling count done strobe indication
0	VCO_TUNE_DONE	R	0h	VCO TUNE Done (All Banks for given iterations) indication - active High till reset or next kickoff

5.3.5.3 CLK_APLL_STATUS_REG2 Register (Offset = 8h) [Reset = 00000000h]

CLK_APLL_STATUS_REG2 is shown in Table 5-386.

Return to the Summary Table.

Table 5-386. CLK_APLL_STATUS_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Unassigned - no PHY net in Analog 0x 0= Functional Reset
15-0	CLK_APLL_STATUS_REG2	R	0h	Readback based on APLL_DIG_RDB_MUX bits in CLK_CTRL_REG11_APLL 15:12> 0x 6 => CLK_APLL_STATUS_REG2 15:0> => fcw_counter_expected 0x 5 => CLK_APLL_STATUS_REG2 15:0> => fcw_counter 0x 4 => CLK_APLL_STATUS_REG2 15:0> => fcw_error 0x 3 => CLK_APLL_STATUS_REG2 15:0> => fcw_error_acc 0x 2 => CLK_APLL_STATUS_REG2 15:0> => beta_scale_selected_val 0x 1 => CLK_APLL_STATUS_REG2 15:0> => < 13:8>bank_code_err < 4:0>bank_code_iter_final 0x 0 => CLK_APLL_STATUS_REG2 15:0> => < 13:10>ufine_bank_code_apply < 9:0>fine_bank_code_apply

5.3.5.4 CLK_CTRL_REG1_APLL Register (Offset = Ch) [Reset = 00000003h]

CLK_CTRL_REG1_APLL is shown in Table 5-387.

Return to the Summary Table.

Table 5-387. CLK_CTRL_REG1_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset

Table 5-387. CLK_CTRL_REG1_APLL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	EN_APLL_BYPASS	R/W	0h	Enable APLL Bypass CLK from CLKM 0 = CLK ADC 400MHz from APLL 1 = CLK ADC 400MHz from CLKM 0x 0 = Functional Reset
24	EN_CLK_DIV4_OUT	R/W	0h	Enable 1000 MHz Clock to APLL DIG TUNE 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
23	EN_CLK_SYNTH	R/W	0h	Enable 400MHz Clock to SYNTH 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
22	EN_CLK_APLL_DIG_800M	R/W	0h	Enable 800MHz Digital Output Clock 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
21	EN_CLK_APLL_DIG_400M	R/W	0h	Enable 400MHz Digital Output Clock 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
20	EN_CLK_ADC	R/W	0h	Enable 400MHz Clock to ADC 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
19	EN_TREE_SYNTHADC	R/W	0h	Enable SYNTH and ADC Root Clock Tree 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
18-17	NDIV_APLL_BURNIN	R/W	0h	Feedback Clock Divider Burn-in Control 00 = Normal operation (divide by 1) 01 = Normal operation (divide by 1) 10 = Burn-in 20 MHz Refclk (divide by 2) 11 = Burn-in 10 MHz Refclk (divide by 4) 0x 0 = Functional Reset
16-10	NDIV_APLL	R/W	0h	APLL Feedback divide ratio - If EN_APLL_PFDCP_DIV_7STG = 0, bit <6> does nothing and 1st stage operates as a 6-stage DIV23 divider. Calculate DIV as 2^6 + < 5:0>. Example for 40 MHz XTAL: 4 GHz input and 40 MHz output is %100. Program 100- 64 = 36 to < 5:0>. - If EN_APLL_PFDCP_DIV7_STG = 1, bit <6> is valid adn the 1st stage operates as a 7-stage DIV23 divider for XTAL frequencies below 38.4 MHz. For example, a 25 MHz XTAL: 4 GHz input and 25 MHz output is %160. Program 160- 128 = 32 to < 6:0>
9	EN_APLL_FILTER	R/W	0h	Enable APLL Active Loop Filter 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
8	EN_APLL_VCO	R/W	0h	Enable APLL VCO 0 = Disabled 1 = Enabled 0x 0 = Functional Reset

Table 5-387. CLK_CTRL_REG1_APLL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	EN_APLL_PFDCP_DIV_7STG	R/W	0h	Enable APLL Feedback Divider 7th Stage For XTALs 25 and 26 MHz only. EN_APLL_PFDCP_DIV must also be enabled. 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
6	EN_APLL_PFDCP_DIV	R/W	0h	Enable APLL Feedback Divider 0 = Disabled 1 = Enabled 0x 0 = Functional Reset
5-2	EN_APLL_CP_3_to_0	R/W	0h	Enable APLL CP <0> enable CP <1> Enable DN switches <2> Enable UP switches <3> SPARE 0x 0 = Functional Reset
1	RESET_SW	R/W	1h	Reset Loop Filter integrator 0 = Closed Loop 1 = Open loop 0x 1 = Functional Reset
0	RESET_APLL	R/W	1h	Reset APLL 1 = RESET 0x 1 = Functional Reset

5.3.5.5 CLK_CTRL_REG10_APLL Register (Offset = 10h) [Reset = 00000000h]

CLK_CTRL_REG10_APLL is shown in Table 5-388.

Return to the Summary Table.

Table 5-388. CLK_CTRL_REG10_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FRAC_IN_LSB_15_to_0	R/W	0h	FRAC_IN_LSB_15_to_0 LSB part of fractional input 0x 0 = Functional Reset
15	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
14	REF_DIV_EN	R/W	0h	ref_div_en Enable REF CLK division for FLL counter, internal FSM works on REF CLK 0x 0 = Functional Reset
13-9	REF_DIV	R/W	0h	ref_div REF CLK division factor for FLL 0x 0 = Functional Reset
8-4	COARSE_BANK_CODE_OVR_IN	R/W	0h	coarse_bank_code_ovr_in Override value for the Coarse (PVT) Bank code to be used if bank_code_ovr_en is set 0x 0 = Functional Reset
3-0	UFINE_BANK_CODE_OVR_IN	R/W	0h	ufine_bank_code_ovr_in Override value for the UFINE Bank code to be used if bank_code_ovr_en is set 0x 0 = Functional Reset

5.3.5.6 CLK_CTRL_REG11_APLL Register (Offset = 14h) [Reset = 0C800800h]

CLK_CTRL_REG11_APLL is shown in Table 5-389.

Return to the Summary Table.

Table 5-389. CLK_CTRL_REG11_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	APLL_DIG_MDIV_IN	R/W	19h	MDIV Value to program the final VCO freq VCO freq = $2*FREF*(4*MDIV+SDIV+(FRAC/2^{20}))$ 0x 19 = Functional Reset

ADVANCE INFORMATION

Table 5-389. CLK_CTRL_REG11_APLL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-20	APLL_DIG_SDIV_IN	R/W	0h	SDIV Value to program the final VCO freq VCO freq = 2*FREF*(4*MDIV+SDIV+(FRAC/2^{20})) 0x 0 = Functional Reset
19-16	APLL_DIG_FRAC_IN_MSB	R/W	0h	MSB part of fractional input 0x 0 = Functional Reset
15-12	APLL_DIG_RDB_MUX	R/W	0h	rdb_mux Read back controls 0x 6 => CLK_APLL_STATUS_REG2 15:0> => fcw_counter_expected 0x 5 => CLK_APLL_STATUS_REG2 15:0> => fcw_counter 0x 4 => CLK_APLL_STATUS_REG2 15:0> => fcw_error 0x 3 => CLK_APLL_STATUS_REG2 15:0> => fcw_error_acc 0x 2 => CLK_APLL_STATUS_REG2 15:0> => beta_scale_selected_val 0x 1 => CLK_APLL_STATUS_REG2 15:0> => 13:8>bank_code_err 4:0>bank_code_iter_final 0x 0 => CLK_APLL_STATUS_REG2 15:0> => 13:10>ufine_bank_code_apply 9:0>fine_bank_code_apply 0x 0 = Functional Reset
11-10	APLL_VCO_START_DELAY	R/W	2h	controls the delay between synth_pll_kickoff and delayed pll_kickoff(when the mid coded gets apllied to the VCO) 00 :64 clock cycles of ref_clk 01:128 clock cycles of ref_clk 10:16 clock cycles of ref_clk 11 :32 clock cycles of ref_clk 0x 2 = Functional Reset
9	APLL_DIG_OPEN_LOOP	R/W	0h	UNUSED 0x 0 = Functional Reset
8	APLL_DIG_ITER_SINGLE_STEP	R/W	0h	UNUSED 0x 0 = Functional Reset
7	APLL_DIG_SD_ORDER	R/W	0h	Sigma delta order, not needed in 64XX 0x 0 = Functional Reset
6	APLL_DIG_SD_EN	R/W	0h	Sigma delta enable, not needed in 64XX 0x 0 = Functional Reset
5	SDIV_FRAC_ASYNC_LOAD	R/W	0h	A toggle of this signal loads new SDIV and Fraction after digital tune, must be stable at least two sdm clock cycles. Added for Osprey support of PLL Relock, not needed in 64XX 0x 0 = Functional Reset
4	APLL_SD_CLK_EN	R/W	0h	APLL DIG SD CLK ENABLE 0 = Sigma delta CLK DISABLE 1 = Sigma delta CLK Enable Not needed in 64XX 0x 0 = Functional Reset
3	APLL_DIG_KICKOFF	R/W	0h	DIG KICKOFF 0 = DISABLE MODE 1 = KICK OFF OR ENABLE VCO CAP TUNE 0x 0 = Functional Reset
2	APLL_DIG_PRESCAL_CLK_EN	R/W	0h	DIG PRESCALER CLK ENABLE 0 = DISABLER PRESCALER CLK 1 = ENABLE PRESCALER CLK 0x 0 = Functional Reset
1	APLL_DIG_REF_CLK_EN	R/W	0h	DIG REF CLK EN 0 = DISABLE REF CLK 1 = ENABLE REF CLK 0x 0 = Functional Reset

Table 5-389. CLK_CTRL_REG11_APLL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	APLL_DIG_RESETZ	R/W	0h	APLL DIG RESETZ 0 = RESET MODE 1 = FUNCTIONAL MODE 0x 0 = Functional Reset

5.3.5.7 CLK_CTRL_REG2_APLL Register (Offset = 18h) [Reset = 00040000h]

CLK_CTRL_REG2_APLL is shown in Table 5-390.

Return to the Summary Table.

Table 5-390. CLK_CTRL_REG2_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
20	EN_APLL_VCONT_MUX	R/W	0h	Controls APLL VCTRL voltage onto TESTMUX (PAD_TEST_O_ANAP). This output is driven by the loop filter. 0'b 0 = Disconnected from TESTMUX 0'b 1 = Connected to TESTMUX
19-16	ICTRL_LF_AMP	R/W	4h	Current control for Loop Filter Amp '0000' - No Current '0001' - Max Current (6mA) '0100' - 1.6mA '0110' - 1.2mA '1111' - Min Current (500uA) 0x 4 = Functional Reset
15-8	APLL_CP_GAIN_PMOS	R/W	0h	CP Gain for DN '0x00' - No Current '0x01' - 125uA 0x 02 - 62uA 0x 04 -125uA 0x 08 - 250uA 0x 10 - 500uA 0x 20 - 1mA < 7:6> not used 0x 0 = Functional Reset
7-0	APLL_CP_GAIN_NMOS	R/W	0h	CP Gain for DN '0x00' - No Current '0x01' - 125uA 0x 02 - 62uA 0x 04 -125uA 0x 08 - 250uA 0x 10 - 500uA 0x 20 - 1mA < 7:6> not used 0x 0 = Functional Reset

5.3.5.8 CLK_CTRL_REG2_LDO_CLKTOP Register (Offset = 1Ch) [Reset = 00000086h]

CLK_CTRL_REG2_LDO_CLKTOP is shown in Table 5-391.

Return to the Summary Table.

Table 5-391. CLK_CTRL_REG2_LDO_CLKTOP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	SPARE1	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
21-18	APLL_CP_LDO_BIST_CTRL	R/W	0h	<18> : VSSA <19>: VIN_SENSE <20>: VOUT SENSE <21> No Connect 0x 0 = Functional Reset
17	APLL_CP_LDO_EN_TEST	R/W	0h	EN TEST Mode 0x 0 = Functional Reset

Table 5-391. CLK_CTRL_REG2_LDO_CLKTOP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	APLL_CPLDO_VTHRESHOLD	R/W	0h	VTHRESHOLD enable = 'H' when LDO is OFF, VTHOLD should be made 'L' with LDO Enable turning 'H' for 30us till LDO output comes up them made 'H' Back. 0x 0 = Functional Reset
15-12	APLL_CPLDO_TEST_MUX_CTRL	R/W	0h	Test MUX CNTRL 0b 0001 = Ibias 0b 0010 = LDOIN 0b 0100 = VOUT 0b 1001 = GND 0x 00 = Functional Reset
11	APLL_CP_LDO_EA_CTRL	R/W	0h	Used in EA block s start-up amp L = Short out 20k degen of current source which pumps current into EA cap H = Enable the 20k degen of current source which pumps current into EA cap 0x 0 = Functional Reset
10	APLL_CP_LDO_EN_BYPASS_MODE	R/W	0h	EN_BYPASS_MODE Bypass LDO with external supply H = Bypass, L = normal operation 0x 0 = Functional Reset
9	APLL_CP_LDO_EN_STARTUP_MODE_2	R/W	0h	EN_STARTUPMODE_2 Enable fast start mode 2 (This is a static bit, It doesn t need to toggle) H = burn 10uA extra current in error amp L = burn only 2.5uA extra current in error amp 0x 0 = Functional Reset
8	APLL_CP_LDO_EN_PMOS_PDN	R/W	0h	EN_PMOSPDN Enable pull down path in buffer to reduce overshoots in Vout L = Enable PD, H = Disable PD 0x 0 = Functional Reset
7	APLL_CP_LDO_EN_HP_MODE	R/W	1h	EN_HPMODE Enable High Power mode, i.e. , burn extra tail current in buffer H = Enable HPmode, L = Disable Hpmode 0x 1 = Functional Reset
6	APLL_CP_LDO_EN_INT_LOAD_1	R/W	0h	EN_INTLOAD<1> Enable internal load. Making any bit H enables 50% of the internal load L = 0mA H = 3.5mA (across PVT: 1.4mA 6mA) 0x 0 = Functional Reset
5-1	APLL_CP_LDO_VOUT_CTRL	R/W	3h	VOUT_TRIM< 4:0> LSB approximately 25mV 0x 3 = Functional Reset
0	APLL_CP_LDO_EN_INT_LOAD_0	R/W	0h	EN_INTLOAD<0> Enable internal load. Making any bit H enables 50% of the internal load L = 0mA H = 3.5mA (across PVT: 1.4mA 6mA) 0x 0 = Functional Reset

5.3.5.9 CLK_CTRL_REG3_APLL Register (Offset = 20h) [Reset = 00011270h]

CLK_CTRL_REG3_APLL is shown in Table 5-392.

Return to the Summary Table.

Table 5-392. CLK_CTRL_REG3_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
16-12	LF_RZ_RCTRL	R/W	11h	Res control for LF Rz 4.5K default 0x 11 = Functional Reset
11-8	LF_CS_CTRL	R/W	2h	Cap control for LF CS <3> - not used '000' - 1.5pF '001' -1.5pF + 1.75pF '010' - 1.5pF+3.5pF '100' 1.5pF+7pF 0x 2 = Functional Reset

Table 5-392. CLK_CTRL_REG3_APLL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-4	SEL_VCTRL	R/W	7h	VCO Calibration DAC Control One-hot control to drive the following voltage to the VCO input. Please refer to the architecture specification for temperature calibration correlation. SEL_VCTRL Vout (V) 0x0 0.24 0x1 0.30 0xn 0.24+ n*0.06 0xF 1.14 0x 7 = Functional Reset
3-2	REF_CP_CTRL	R/W	0h	Res control for Loop filter positive node VCM '00' not connected (default for off state) '01' 12KOhm (default for on state) '10' 8KOhm '11' 12K||8KOhm = 4.8K 0x 0 = Functional Reset
1	VCO_CAL_DAC_EN	R/W	0h	Enable VCO Calibration DAC Enables DAC to drive the APLL VCO control voltage for temperature calibration. 0 = Normal closed-loop operation (DAC is off = High impedance) 1 = Drive VCO control voltage with programmed DAC value 0x 0 = Functional Reset
0	TST_BUFEN	R/W	0h	Testmux Buffer Enable for Vcontrol 0 = Normal operation (high impedance on internal analog test bus - PAD_TEST_O_ANAP) 1 = Drives buffered version of VCO input control voltage from the PFDCP on internal analog test bus - PAD_TEST_O_ANAP 0x 0 = Functional Reset

5.3.5.10 CLK_CTRL_REG3_LDO_CLKTOP Register (Offset = 24h) [Reset = 00600043h]

CLK_CTRL_REG3_LDO_CLKTOP is shown in Table 5-393.

Return to the Summary Table.

Table 5-393. CLK_CTRL_REG3_LDO_CLKTOP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
23-20	BISTMUX_CTRL	R/W	6h	BIST MUX CONTROL 0110 = 0.6*VLDO Output 0x 6 = Functional Reset
19-16	TESTMUX_CTRL	R/W	0h	APLL VCO LDO TEST CONTROL (ONE HOT) Analog MUX enables to test output port 0000 = HI-Z Output 0001 = 0.5*VDD18 0010 = 0.6*VLDO Output 0100 = VSSA 1000 = VFB (0.9V) WARNING: Enabling more than one bit may damage the device 0x 0 = Functional Reset
15-13	TLOAD_CTRL	R/W	0h	APLL VCO LDO TLOAD CONTROL Value should be 0x0 during boot sequence to ensure stability while unloaded, then 0x1 to turn off all current loading after oscillator is enabled to reduce power and extend reliability. Iload=undefined*24mA+undefined*16mA+!undefined*8mA 0b 001 = no current load 0b 000 = 8mA load 0b 010 = 16mA load 0b 100 = 24mA load 0x 0 = Functional Reset
12	ENABLE_PMOS_PULLDOWN	R/W	0h	APLL VCO LDO PMOS PULL DOWN ENABLE 0 = Slicer LDO PMOS Pull Down disabled 1 = Slicer LDO PMOS Pull Down enabled 0x 0 = Functional Reset

Table 5-393. CLK_CTRL_REG3_LDO_CLKTOP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SCPRT_IBIAS_CTRL	R/W	0h	APLL VCO LDO SHORT CKT PROTECTION IBIAS CONTROL 0 = Nominal short circuit bias with nominal short circuit current limit 1 = 2X Nominal short circuit bias with higher short circuit current limit 0x 0 = Functional Reset
10-8	LDO_BW_CTRL	R/W	0h	BW CONTROL FOR APLL VCO LDO 0x 0 - Highest BW 0x 1 - Mid BW 0x 2 - Lowest BW 0x 0 = Functional Reset
7	EN_BYPASS	R/W	0h	APLL VCO LDO BYPASS ENABLE 0 = APLL VCO LDO in normal mode 1 = APLL VCO LDO Bypassed with external voltage 0x 0 = Functional Reset
6	EN_SHRT_CKT	R/W	1h	APLL VCO LDO SHORT CKT PROTECTION DISABLE 0 = APLL VCO LDO Short Ckt Protection Enabled 1 = APLL VCO LDO Short Ckt Protection Disabled 0x 1 = Functional Reset
5	EN_TEST_MODE	R/W	0h	APLL VCO LDO TEST MODE ENABLE 0 = APLL VCO LDO TEST MODE Disabled 1 = APLL VCO LDO TEST MODE Enabled 0x 0 = Functional Reset
4-0	LDO_VOUT_CTRL	R/W	3h	APLL VCO LDO VOUT TRIM 0bXXXX 0 = 1.238 0bXXX 01 = 1.350 0bXX 011 = 1.463 0bX 0111 = 1.519 0bX 1111 = 1.575 0x 03 = Functional Reset

5.3.5.11 CLK_CTRL_REG4_APLL Register (Offset = 28h) [Reset = 00000000h]

CLK_CTRL_REG4_APLL is shown in Table 5-394.

Return to the Summary Table.

Table 5-394. CLK_CTRL_REG4_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset

5.3.5.12 CLK_CTRL_REG4_LDO_CLKTOP Register (Offset = 2Ch) [Reset = 0040071Dh]

CLK_CTRL_REG4_LDO_CLKTOP is shown in Table 5-395.

Return to the Summary Table.

Table 5-395. CLK_CTRL_REG4_LDO_CLKTOP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset

Table 5-395. CLK_CTRL_REG4_LDO_CLKTOP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-20	BISTMUX_CTRL_LDO_APLL	R/W	4h	CLKTOP IOBUF APLL LDO BIST CONTROL (ONE HOT) Analog MUX enables to BIST output port 0000 = HI-Z Output 0001 = VBG_0P9*10/ 9 =1.0 V 0010 = VDD18*0. 5 = 0.9V 0100 = VLDO Output * 0.6 1000 = Floating WARNING: Enabling more than one bit may damage the device 0x 4 = Functional Reset
19-16	TESTMUX_CTRL_LDO_APLL	R/W	0h	CLKTOP IOBUF APLL LDO TEST CONTROL (ONE HOT) Analog MUX enables to test output port 0000 = HI-Z Output 0001 = 0.6 * VLDO_OUT 0010 = VDD18*0. 5 = 0.9V 0100 = VSSA 1000 = LDO Test Current (12.5uA) WARNING: Enabling more than one bit may damage the device 0x 0 = Functional Reset
15-13	TLOAD_CTRL	R/W	0h	CLK_TOP IOBUF APLL/ROUTE LDO TLOAD CONTROL Value should be 0x0 during boot sequence to ensure stability while unloaded, then 0x1 to turn off all current loading after oscillator is enabled to reduce power and extend reliability. Iload=undefined*24mA+undefined*16mA+!undefined*8mA 0b 001 = no current load 0b 000 = 8mA load 0b 010 = 16mA load 0b 100 = 24mA load 0x 0 = Functional Reset
12	ENABLE_PMOS_PULLDOWN	R/W	0h	CLK_TOP IOBUF APLL/ROUTE LDO PMOS PULL DOWN ENABLE 0 = Slicer LDO PMOS Pull Down disabled 1 = Slicer LDO PMOS Pull Down enabled 0x 0 = Functional Reset
11	SCPRT_IBIAS_CTRL	R/W	0h	CLK_TOP IOBUF APLL/ROUTE LDO SHORT CKT PROTECTION IBIAS CONTROL 0 = Nominal short circuit bias with nominal short circuit current limit 1 = 2X Nominal short circuit bias with higher short circuit current limit 0x 0 = Functional Reset
10-8	LDO_BW_CTRL	R/W	7h	CLK_TOP IOBUF APLL/ROUTE LDO BANDWIDTH CONTROL Control the bias current in the fast loop buffer of the SLICER LDO, in steps of 10uA 101 - 30uA 111 - 50uA (default) 010 - 100uA 0x 7 = Functional Reset
7	EN_BYPASS	R/W	0h	CLK_TOP IOBUF APLL/ROUTE LDO BYPASS ENABLE 0 = Slicer LDO in normal mode 1 = Slicer LDO Bypassed with external voltage 0x 0 = Functional Reset
6	EN_SHRT_CKT	R/W	0h	CLK_TOP IOBUF APLL/ROUTE LDO SHORT CKT PROTECTION ENABLE 0 = Slicer LDO Short Ckt Protection Disabled 1 = Slicer LDO Short Ckt Protection Enabled 0x 0 = Functional Reset
5	EN_TEST_MODE	R/W	0h	CLK_TOP IOBUF APLL/ROUTE LDO TEST MODE ENABLE 0 = Slicer LDO TEST MODE Disabled 1 = Slicer LDO TEST MODE Enabled 0x 0 = Functional Reset

ADVANCE INFORMATION

Table 5-395. CLK_CTRL_REG4_LDO_CLKTOP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ENZ_LOW_BW_CAP	R/W	1h	CLK_TOP IOBUF APLL/ROUTE LDO LOW BW MODE ENABLE 0 = Slicer LDO Low BW mode Disabled 1 = Slicer LDO Low BW mode Enabled 0x 1 = Functional Reset
3-0	LDO_VOUT_CTRL	R/W	Dh	CLK_TOP IOBUF APLL/ROUTE LDO VOUT TRIM Trim the LDO output voltage, in steps of 25mV 0000 - 1.40V 0001 - 1.375V 0010 - 1.35V 0011 - 1.325V 0100 - 1.30V 0101 - 1.275V 0110 - 1.25V 0111 - 1.225V 1000 - 1.60V 1001 - 1.575V 1010 - 1.55V 1011 - 1.525V 1100 - 1.50V 1101 - 1.475V 1110 - 1.45V 1111 - 1.425V 0xD = Functional Reset

5.3.5.13 CLK_CTRL_REG5_APLL Register (Offset = 30h) [Reset = 00300000h]

CLK_CTRL_REG5_APLL is shown in Table 5-396.

Return to the Summary Table.

Table 5-396. CLK_CTRL_REG5_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
23-18	RTRIM_VCO_APLL	R/W	Ch	VCO RTrim for Gm Stage Bias 0xC= Functional Reset
17-10	APLL_FINE_CTRIM_OVR	R/W	0h	Override value for VCO FINE CTRIM 0x 0 = functional reset
9-6	APLL_UFINE_CTRIM_OVR	R/W	0h	Override value for VCO UFINE CTRIM 0x 0 = functional reset
5-1	APLL_COARSE_CTRIM_OVR	R/W	0h	Override value for VCO COARSE CTRIM 0x 0 = functional reset
0	EN_APLL_CTRIM_OVERRIDE	R/W	0h	Enable override/bypassing of APLL DIG TUNE block and directly feed APLL CTRIM 0 = APLL DIG TUNE drives CTRIM 1 = Use override values below 0x 0 = functional reset

5.3.5.14 CLK_CTRL_REG6_APLL Register (Offset = 34h) [Reset = 00501028h]

CLK_CTRL_REG6_APLL is shown in Table 5-397.

Return to the Summary Table.

Table 5-397. CLK_CTRL_REG6_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	APLL_BETA_COARSE_IN	R/W	50h	Beta value for Coarse (PVT) bank APLL VCO cap tuning 0x 50 = Functional Reset
15	APLL_BANK_SKIP_IN_COARSE	R/W	0h	Set this bit high to skip Coarse (PVT) bank during tuning. The Mid value configured will apply as the Coarse (PVT) Code 0x 0 = Functional Reset

Table 5-397. CLK_CTRL_REG6_APLL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-13	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
12-8	APLL_MID_VAL_COARSE	R/W	10h	start value for APLL VCO Coarse (PVT) Cap code 0x 10 = Functional Reset
7-0	APLL_ACC_WIN_SIZE_COARSE	R/W	28h	Number of FREFs over which the FCW should be integrated for error computation for PVT tuning 0x 28 = Functional Reset

5.3.5.15 CLK_CTRL_REG7_APLL Register (Offset = 38h) [Reset = 00700482h]

CLK_CTRL_REG7_APLL is shown in Table 5-398.

Return to the Summary Table.

Table 5-398. CLK_CTRL_REG7_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	APLL_BETA_FINE_IN	R/W	70h	Beta value for FINE (AB) bank APLL VCO cap tuning 0x 70 = Functional Reset
15	APLL_BANK_SKIP_IN_FINE	R/W	0h	Set this bit high to skip Fine (AB) bank during tuning. The Mid value configured will apply as the Fine (AB) Code 0x 0 = Functional Reset
14-12	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
11-8	APLL_MID_VAL_FINE	R/W	4h	start value for APLL VCO Fine (AB) Cap code 0x 4 = Functional Reset
7-0	APLL_ACC_WIN_SIZE_FINE	R/W	82h	Number of FREFs over which the FCW should be integrated for error computation for Fine (AB- Acquisition Bank) tuning 0x 82 = Functional Reset

5.3.5.16 CLK_CTRL_REG8_APLL Register (Offset = 3Ch) [Reset = 009002FEh]

CLK_CTRL_REG8_APLL is shown in Table 5-399.

Return to the Summary Table.

Table 5-399. CLK_CTRL_REG8_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	APLL_BETA_UFINE_IN	R/W	90h	Beta value for Ultra FINE bank APLL VCO cap tuning 0x 90 = Functional Reset
15	APLL_BANK_SKIP_IN_UFINE	R/W	0h	Set this bit high to skip Ultra FINE bank during tuning. The Mid value configured will apply as the AB Code 0x 0 = Functional Reset
14-11	SPARE0	R/W	0h	Spare Reads return 0x0 and writes have no effect. 0x 0 = Functional Reset
10-8	APLL_MID_VAL_UFINE	R/W	2h	start value for APLL VCO Ultra FINE Cap code 0x 2 = Functional Reset
7-0	APLL_ACC_WIN_SIZE_UFINE	R/W	FEh	Number of FREFs over which the FCW should be integrated for error computation for ULTRA FINE cap bank tuning 0x 0 = Functional Reset

5.3.5.17 CLK_CTRL_REG9_APLL Register (Offset = 40h) [Reset = 10001FF8h]

CLK_CTRL_REG9_APLL is shown in Table 5-400.

Return to the Summary Table.

Table 5-400. CLK_CTRL_REG9_APLL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	APLL_DIG_BANK_CODE_OVR_EN	R/W	0h	bank_code_ovr_en Override computed bank code values Set this bit to use override bank code values instead of what is computed every iteration. Can be used for debug mode or FW workarounds 0x 0 = Functional Reset
30-26	ITER_EXIT_UFINE_COUNT_IN	R/W	4h	iter_exit_ufine_count_in Number of counts after which the iterations for UFINE bank should stop and signal end of tuning for FINE bank for signalling end of tune indication 0x 4 = Functional Reset
25-16	FINE_BANK_CODE_OVR_IN	R/W	0h	fine_bank_code_ovr_in Override value for the FINE (AB) Bank code to be used if bank_code_ovr_en is set 0x 0 = Functional Reset
15-14	APLL_DIG_MODE	R/W	0h	pll_mode Defines if the PLL should work in Auto Mode or Stepping Mode 00: Full auto mode for Bank switching as well as on every iteration within the bank tuning 01: Iteration stepping is auto, but Bank switching is Manual. After every bank tuning, interrupt is generated and tuning is paused. Next bank tuning resumes on synth_kickoff retrigger 10: Bank switching is auto, but Iteration stepping is Manual. After every iteration, interrupt is generated and tuning is paused and resumes on iteration_single_step_in trigger 11: Both Bank switching and Iteration stepping are manual. Interrupt is generated on every iteration step as well as bank switch and resume on iteration_single_step_in and synth_kickoff triggers respectively 0x 0 = Functional Reset
13-9	ITER_EXIT_FINE_COUNT_IN	R/W	Fh	iter_exit_fine_count_in Number of counts after which the iterations for AB bank should stop and signal end of tuning for FINE (AB) bank for bank switching to UFINE 0xF = Functional Reset
8-4	ITER_EXIT_COARSE_COUNT_IN	R/W	1Fh	iter_exit_coarse_count_in Number of counts after which the iterations for PVT bank should stop and signal end of tuning for coarse (PVT) bank for bank switching to FINE (AB) 0x1F = Functional Reset
3-0	APLL_VCO_SETTLING_COUNT_IN	R/W	8h	vco_settling_count_in Number of FREF clocks to wait for VCO to settle after every bank code update till error accumulation count starts 0x 8 = Functional Reset

5.3.5.18 CLK_MDLL_REG1 Register (Offset = 44h) [Reset = 88A02A08h]

CLK_MDLL_REG1 is shown in Table 5-401.

Return to the Summary Table.

Table 5-401. CLK_MDLL_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	MDLL_RTRIM	R/W	4h	RTRIM for Delay cell comparator 0x 4 = Functional Reset
28	MDLL_C_CTRL2	R/W	0h	Delay cell cap 2 '0' higher delay '1' lowest delay 0x 0 = Functional Reset
27-25	MDLL_RTRIM_2	R/W	4h	RTRIM2 for Delay cell comparator 0x 4 = Functional Reset
24	MDLL_C_CTRL3	R/W	0h	Delay cell cap 3 '0' higher delay '1' lowest delay 0x 0 = Functional Reset
23	MDLL_HIGH_POWER_2	R/W	1h	increase current in delay cell comparator 0x 1 = Functional Reset

Table 5-401. CLK_MDLL_REG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-21	MDLL_C_CTRL	R/W	1h	Delay cell cap '00' higher delay '11' lowest delay 0x 1 = Functional Reset
20	MDLL_OPEN_LOOP	R/W	0h	To open the DLL loop 1 = Open loop for DLL 0x 0 = Functional Reset
19	MDLL_EXTN_VTUNE	R/W	0h	To send a VTUNE to delay cell for debug mode, from a R ladder 0x 0 = Functional Reset
18-16	MDLL_SEL_VCTRL_2_to_0	R/W	0h	Select VTUNE voltage from R ladder '000' highest voltage '111' lowest voltage 0x 0 = Functional Reset
15	MDLL_EN	R/W	0h	1' Enable DLL '0' Disable DLL 0x 0 = Functional Reset
14	MDLL_HIGH_POWER	R/W	0h	1' Enable High Power '0' Disable High Power 0x 0 = Functional Reset
13	MDLL_RESET	R/W	1h	1' Reset DLL '0' Disable RESET DLL 0x 1 = Functional Reset
12	MDLL_EN_IQMM_TEST	R/W	0h	1' Enables DLL Output for IQMM test 0x 0 = Functional Reset
11-10	MDLL_PFD_DELAY_1_to_0	R/W	2h	00' - One Nand delay '01' - One Nand delay '10' - One NAND and two inverter delay '11' - Max delay 0x 2 = Functional Reset
9-8	MDLL_CP_GAIN_1_to_0	R/W	2h	CP Gain trim setting 00' - 40uA '10' and '01' - 53uA '11' - 60uA 0x 2 = Functional Reset
7-6	MDLL_LOOP_FILT_1_to_0	R/W	0h	LF Cap control '00' - 3pF (Max cap) '01' and '10' 2pF '11' - 1pF 0x 0 = Functional Reset
5-1	MDLL_REF_DIV_4_to_0	R/W	4h	MDLL Multiplication Factor 0x 2 = MDLL_CLK = 2 X XTAL CLK 0x 4 = MDLL_CLK = 4 X XTAL CLK 0x 4 = Functional Reset
0	MDLL_EN_POWER_SW	R/W	0h	1' - enables DLL supply power sw 0x 0 = Functional Reset

ADVANCE INFORMATION

5.3.5.19 CLK_STATUS_REG Register (Offset = 48h) [Reset = 00000000h]

CLK_STATUS_REG is shown in Table 5-402.

Return to the Summary Table.

Table 5-402. CLK_STATUS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Unassigned - no PHY net in Analog 0x 0= Functional Reset
7	MULT_CHAIN_LDO_SC_OUT	R	0h	Multiplier chain LDO short circuit protection INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
6	CLKTOP_IOBUF_ROUTE_LDO_SC_OUT	R	0h	CLKTOP IOBUF ROUTE LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
5	SYNTH_DIV_LDO_SC_OUT	R	0h	SYNTH DIV LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
4	SYNTH_VCO_LDO_SC_OUT	R	0h	SYNTH VCO LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected

Table 5-402. CLK_STATUS_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	SDM_LDO_SC_OUT	R	0h	SDM LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
2	CLKTOP_IOBUF_APLL_LDO_SC_OUT	R	0h	CLKTOP IOBUF APLL LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
1	APLL_VCO_LDO_SC_OUT	R	0h	APLL VCO LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
0	SLICER_LDO_SC_OUT	R	0h	SLICER LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected

5.3.5.20 CLK_XTAL_X2_REG1 Register (Offset = 4Ch) [Reset = 86182AA8h]

CLK_XTAL_X2_REG1 is shown in Table 5-403.

Return to the Summary Table.

Table 5-403. CLK_XTAL_X2_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DCC3_OUT_MODE_SEL	R/W	1h	Select or bypass the Output DCC (DCC3): 0 - Bypass DCC3 1 - Engage DCC3 0x 1 = Functional Reset
30-28	X2_DCC3_OPAMP_RLOAD_CTRL	R/W	0h	Control DCC3 Opamp Load Resistor value: 000 - 400k? 001 - 500k? 010 - 600k? 011 - 700k? 1XX - 800k? 0x 0 = Functional Reset
27-26	X2_DCC3_CAP_CTRL	R/W	1h	Control DCC3 Cap value: 00 - 130f 01 - 260f (default) 10 - 390f 11 - 520f 0x 1 = Functional Reset
25	EN_OUTPUT_DCC3	R/W	1h	Enable the output Duty Cycle Correction stage (DCC3) 0x 1 = Functional Reset
24-22	X2_DCC2_OPAMP_RLOAD_CTRL	R/W	0h	Control DCC2 Opamp Load Resistor value: 000 - 400k? 001 - 500k? 010 - 600k? 011 - 700k? 1XX - 800k? 0x 0 = Functional Reset
21-20	X2_DCC2_CAP_CTRL	R/W	1h	Control DCC2 Cap value: 00 - 130f 01 - 260f (default) 10 - 390f 11 - 520f 0x 1 = Functional Reset
19	RESERVED	R/W	1h	Unused 0x 1 = Functional Reset
18-16	X2_DCC1_OPAMP_RLOAD_CTRL	R/W	0h	Control DCC1 Opamp Load Resistor value: 0000 - 800k? 0001 - 700k? 001X - 600k? (default) 01XX - 500k? 1XXX - 400k? (default for stable startup) 0x 8 = Functional Reset
15	X2_EN_DOUBLER	R/W	0h	Enable the CLK_XTAL_DOUBLER 0x 0 = Functional Reset

Table 5-403. CLK_XTAL_X2_REG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-13	X2_DCC1_CAP_CTRL	R/W	1h	Control DCC1 Cap value: 00 - 130f 01 - 260f (default) 10 - 390f 11 - 520f 0x 1 = Functional Reset
12-11	X2_DCC3_OPAMP_IREF_CTRL	R/W	1h	Control DCC3 Amp reference current: 00 - 2.5uA 01 - 5uA (default) 10 - 7.5uA 11 - 10uA 0x 1 = Functional Reset
10-9	X2_DCC2_OPAMP_IREF_CTRL	R/W	1h	Control DCC2 Amp reference current: 00 - 2.5uA 01 - 5uA (default) 10 - 7.5uA 11 - 10uA 0x 1= Functional Reset
8-7	X2_DCC1_OPAMP_IREF_CTRL	R/W	1h	Control DCC1 Amp reference current: 00 - 2.5uA 01 - 5uA (default) 10 - 7.5uA 11 - 10uA 0x 1 = Functional Reset
6-3	X2_RTRIM	R/W	5h	Rtrim bits (parallel) for Pulse Gen resistor Nom - 0101 Weak - 0001 Strong - 1101 0x 5 = Functional Reset
2	X2_MODE_SEL	R/W	0h	Select the Duty Cycle Correction Mode: 0 - Bypass DCC1 & DCC2 1 - Engage DCC1 & DCC2 0x 0 = Functional Reset
1	X2_EN_INPUT_DCC	R/W	0h	Enable the input Duty Cycle Correction stages (DCC1 & DCC2) 0x 0 = Functional Reset
0	PULSE_WIDTH_CTRL	R/W	0h	Increase PulseGen Cap by 30%, to increase pulse width by 30% 0x 0 = Functional Reset

ADVANCE INFORMATION

5.3.5.21 REFSYS_CTRL_REG0_LOWV Register (Offset = 50h) [Reset = 00000000h]

REFSYS_CTRL_REG0_LOWV is shown in Table 5-404.

Return to the Summary Table.

Table 5-404. REFSYS_CTRL_REG0_LOWV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	Unassigned - no PHY net in Analog 0x 0= Functional Reset
8	SPARE2	R/W	0h	SPARE 0x 0 = Functional Reset
7-0	SPARE1	R/W	0h	SPARE 0x 0 = Functional Reset

5.3.5.22 WU_SPARE_OUT_LOWV Register (Offset = 54h) [Reset = 00000028h]

WU_SPARE_OUT_LOWV is shown in Table 5-405.

Return to the Summary Table.

Table 5-405. WU_SPARE_OUT_LOWV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Unassigned - no PHY net in Analog 0x 0= Functional Reset
7	CORE_UVDET_LOWV	R	0h	UV Detect of Core Supply-Unlatched
6	CORE_OVDET_LOWV	R	0h	OV Detect of Core Supply-Unlatched
5	INT_OSC_CTRL	R	1h	Internal Oscillator Control
4	SUPP_DET_OR_CTRL	R	0h	Supply Detect or control
3	HVMODE	R	1h	HVMODE Status from VMON 1 = 3.3V VIO 0 = 1.8V VIO
2	VDDS18DET	R	0h	Status of 1.8V IO Bias Supply
1	VDDS33DET	R	0h	spare. Internal tie low.
0	SUPP_DET_OVERRIDE	R	0h	spare. Internal tie low.

5.3.5.23 WU_STATUS_REG_LOWV Register (Offset = 58h) [Reset = 00000014h]

WU_STATUS_REG_LOWV is shown in Table 5-406.

Return to the Summary Table.

Table 5-406. WU_STATUS_REG_LOWV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Unassigned - no PHY net in Analog 0x 0= Functional Reset
10-9	SPARE0	R	0h	SPARE
8	SRAM_SC_OUT	R	0h	SRAM LDO SC OUT 1 = short circuit detected 0 = No short circuit
7	DIG_SC_OUT	R	0h	DIG LDO SC OUT 1 = short circuit detected 0 = No short circuit
6	SC_OUT_1210	R	0h	1210 LDO SC OUT 1 = short circuit detected 0 = No short circuit
5	SC_OUT_1812	R	0h	1812 LDO SC OUT 1 = short circuit detected 0 = No short circuit
4	HVMODE	R	1h	HVMODE Status from VMON 1 = 3.3V VIO 0 = 1.8V VIO
3	SUPP_OK_IO18	R	0h	Supp Detect output of IO 1.8V 0 = Supply Not detected 1 = Supply Detected
2	SUPP_OK_IO33	R	1h	Supp Detect output of IO 3.3V 0 = Supply Not detected 1 = Supply Detected
1	CORE_UVDET_LAT	R	0h	Latched Value of UV Detect 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
0	CORE_OVDET_LAT	R	0h	Latched Value of OV Detect 0 = OV Detect Not Triggered 1 = OV Detect has Triggered

5.3.5.24 ANALOG_WU_STATUS_REG_POLARITY_INV Register (Offset = 5Ch) [Reset = X]

ANALOG_WU_STATUS_REG_POLARITY_INV is shown in Table 5-407.

Return to the Summary Table.

Table 5-407. ANALOG_WU_STATUS_REG_POLARITY_INV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	X	
10-0	INV_CTRL	R/W	0h	This register decides the polarity of each status bit before providing to the APP_ESM. Each bit controls the respective status bit.

5.3.5.25 ANALOG_CLK_STATUS_REG_POLARITY_INV Register (Offset = 60h) [Reset = X]

ANALOG_CLK_STATUS_REG_POLARITY_INV is shown in Table 5-408.

Return to the Summary Table.

Table 5-408. ANALOG_CLK_STATUS_REG_POLARITY_INV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	INV_CTRL	R/W	0h	This register decides the polarity of each status bit before providing to the APP_ESM. Each bit controls the respective status bit.

5.3.5.26 ANALOG_WU_STATUS_REG_MASK Register (Offset = 64h) [Reset = X]

ANALOG_WU_STATUS_REG_MASK is shown in Table 5-409.

Return to the Summary Table.

Table 5-409. ANALOG_WU_STATUS_REG_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	X	
10-0	MASK	R/W	7FFh	Writing 1'b 1 : Masks the corresponding status bit before generating a group 1 ESM error. 1'b 0 : Unmasks the corresponding status bit before generating a group 1 ESM error.

5.3.5.27 ANALOG_CLK_STATUS_REG_MASK Register (Offset = 68h) [Reset = X]

ANALOG_CLK_STATUS_REG_MASK is shown in Table 5-410.

Return to the Summary Table.

Table 5-410. ANALOG_CLK_STATUS_REG_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	MASK	R/W	FFh	Writing 1'b 1 : Masks the corresponding status bit before generating a group 1 ESM error. 1'b 0 : Unmasks the corresponding status bit before generating a group 1 ESM error.

5.3.5.28 ANALOG_CLK_GOOD_STATUS Register (Offset = 6Ch) [Reset = X]

ANALOG_CLK_GOOD_STATUS is shown in Table 5-411.

Return to the Summary Table.

Table 5-411. ANALOG_CLK_GOOD_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	X	
1	RCOSC32K_GOOD	R	1h	This register gives status of CLK_GOOD_RCOSC32K_LOWV
0	RCOSC10M_GOOD	R	1h	This register gives status of CLK_GOOD_RCOSC10M_LOWV

5.3.5.29 ANALOG_CLK_GOOD_MASK Register (Offset = 70h) [Reset = X]

ANALOG_CLK_GOOD_MASK is shown in Table 5-412.

Return to the Summary Table.

Table 5-412. ANALOG_CLK_GOOD_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	RCOSC32K_GOOD	R/W	1h	Writing 1'b 1 : Masks the corresponding status bit before generating a group 1 ESM error. 1'b 0 : Unmasks the corresponding status bit before generating a group 1 ESM error.
0	RCOSC10M_GOOD	R/W	1h	Writing 1'b 1 : Masks the corresponding status bit before generating a group 1 ESM error. 1'b 0 : Unmasks the corresponding status bit before generating a group 1 ESM error.

5.3.5.30 LOCK0_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0_KICK0 is shown in Table 5-413.

Return to the Summary Table.

- KICK0 component

Table 5-413. LOCK0_KICK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

5.3.5.31 LOCK0_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0_KICK1 is shown in Table 5-414.

Return to the Summary Table.

- KICK1 component

Table 5-414. LOCK0_KICK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

5.3.5.32 intr_raw_status Register (Offset = 1010h) [Reset = X]

intr_raw_status is shown in Table 5-415.

Return to the Summary Table.

Interrupt Raw Status/Set Register

Table 5-415. intr_raw_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

5.3.5.33 intr_enabled_status_clear Register (Offset = 1014h) [Reset = X]

intr_enabled_status_clear is shown in Table 5-416.

Return to the Summary Table.

Interrupt Enabled Status/Clear register

Table 5-416. intr_enabled_status_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

5.3.5.34 intr_enable Register (Offset = 1018h) [Reset = X]

intr_enable is shown in Table 5-417.

Return to the Summary Table.

Interrupt Enable register

Table 5-417. intr_enable Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

5.3.5.35 intr_enable_clear Register (Offset = 101Ch) [Reset = X]

intr_enable_clear is shown in Table 5-418.

Return to the Summary Table.

Interrupt Enable Clear register

Table 5-418. intr_enable_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

5.3.5.36 eoi Register (Offset = 1020h) [Reset = X]

eoi is shown in Table 5-419.

Return to the Summary Table.

EOI register

Table 5-419. eoi Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

5.3.5.37 fault_address Register (Offset = 1024h) [Reset = 00000000h]

fault_address is shown in Table 5-420.

Return to the Summary Table.

Fault Address register

Table 5-420. fault_address Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

5.3.5.38 fault_type_status Register (Offset = 1028h) [Reset = X]

fault_type_status is shown in Table 5-421.

Return to the Summary Table.

Fault Type Status register

Table 5-421. fault_type_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	X	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_ 0000 = Supervisor read fault - priv = 1 dir = 1 dtype != 1 01_ 0000 = Supervisor write fault - priv = 1 dir = 0 00_ 1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_ 0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_ 0010 = User write fault - priv = 0 dir = 0 00_ 0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_ 0000 = No fault

5.3.5.39 fault_attr_status Register (Offset = 102Ch) [Reset = 00000000h]

fault_attr_status is shown in Table 5-422.

Return to the Summary Table.

Fault Attribute Status register

Table 5-422. fault_attr_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

5.3.5.40 fault_clear Register (Offset = 1030h) [Reset = X]

fault_clear is shown in Table 5-423.

Return to the Summary Table.

Fault Clear register

Table 5-423. fault_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	X	

Table 5-423. fault_clear Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

5.3.6 PLLDIG_CTRL Registers

Table 5-424 lists the memory-mapped registers for the PLLDIG_CTRL registers. All register offset addresses not listed in Table 5-424 should be considered as reserved locations and the register contents should not be modified.

Table 5-424. PLLDIG_CTRL Registers

Offset	Acronym	Register Name	Section
0h	PID	PID register	Go
4h	PLLDIG_EN		Go
8h	PLLDIG_MDIV_NDIV		Go
Ch	PLLDIG_CTRL		Go
10h	PLLDIG_MODE_EN		Go
14h	PLLDIG_APLL_SW_DIS_DELAY1		Go
18h	PLLDIG_APLL_SW_DIS_DELAY2		Go
1Ch	PLLDIG_OVERRIDE		Go
20h	PLLDIG_STATUS		Go
24h	FAST_CLK_MUX_POSTDIV		Go
28h	FAST_CLK_STATUS		Go
1008h	LOCK0_KICK0	- KICK0 component	Go
100Ch	LOCK0_KICK1	- KICK1 component	Go
1010h	intr_raw_status	Interrupt Raw Status/Set Register	Go
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	Go
1018h	intr_enable	Interrupt Enable register	Go
101Ch	intr_enable_clear	Interrupt Enable Clear register	Go
1020h	eoi	EOI register	Go
1024h	fault_address	Fault Address register	Go
1028h	fault_type_status	Fault Type Status register	Go
102Ch	fault_attr_status	Fault Attribute Status register	Go
1030h	fault_clear	Fault Clear register	Go

Complex bit access types are encoded to fit into small table cells. Table 5-425 shows the codes that are used for access types in this section.

Table 5-425. PLLDIG_CTRL Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

5.3.6.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in Table 5-426.

Return to the Summary Table.

PID register

Table 5-426. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

5.3.6.2 PLLDIG_EN Register (Offset = 4h) [Reset = X]

PLLDIG_EN is shown in Table 5-427.

Return to the Summary Table.

Table 5-427. PLLDIG_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	X	
18-16	cfg_plldig_lockmon_enable	R/W	0h	PLL DIG lockmon enable 0x 0 = PLL DIG lockmon disbale 0x 7 = PLL DIG lockmon enable
15-11	RESERVED	R/W	X	
10-8	cfg_pll_auto_switch_enable	R/W	0h	PLL DIG and APLL auto switch enable 0x 0 = PLL DIG wont be auto turn off when APLL is enable 0x 7 = PLL DIG will be auto turn off when APLL is enable
7-3	RESERVED	R/W	X	
2-0	cfg_plldig_en	R/W	0h	PLL DIG enable 0x 0 = PLL DIG disable 0x 7 = PLL DIG enable

5.3.6.3 PLLDIG_MDIV_NDIV Register (Offset = 8h) [Reset = X]

PLLDIG_MDIV_NDIV is shown in Table 5-428.

Return to the Summary Table.

Table 5-428. PLLDIG_MDIV_NDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	X	
22-16	cfg_plldig_ndiv	R/W	0h	NDIV value for the PLL DIG Input clock divider settings .NDIV value directly programs the 7-bit pre- divider. Divide value ranges from 2 to 127. NDIV value has to be chosen based on the REF_CLKIN frequency so as to get the internal reference frequency of the PLL closest to 2Mhz
15-9	RESERVED	R/W	X	
8-0	cfg_plldig_mdiv	R/W	0h	MDIV value for the PLL DIG Feedback divider settings. MDIV value directly programs the 9-bit feedback divider. Divide value ranges from 2 to 511. MDIV value has to be chosen to generate the required clock out frequency from the 2Mhz internal PLL reference

5.3.6.4 PLLDIG_CTRL Register (Offset = Ch) [Reset = 00000000h]

PLLDIG_CTRL is shown in Table 5-429.

Return to the Summary Table.

Table 5-429. PLLDIG_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	cfg_plldig_ctrl	R/W	0h	PLL DIG test controls

5.3.6.5 PLLDIG_MODE_EN Register (Offset = 10h) [Reset = X]

PLLDIG_MODE_EN is shown in Table 5-430.

Return to the Summary Table.

Table 5-430. PLLDIG_MODE_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	X	
16	cfg_plldig_lowfreq_mode_en	R/W	1h	PLL DIG high frequency mode operation, Divide by 2 the PLL clock out
15-1	RESERVED	R/W	X	
0	cfg_plldig_highfreq_mode_en	R/W	1h	PLL DIG high frequency mode operation

5.3.6.6 PLLDIG_APLL_SW_DIS_DELAY1 Register (Offset = 14h) [Reset = 0FA00FA0h]

PLLDIG_APLL_SW_DIS_DELAY1 is shown in Table 5-431.

Return to the Summary Table.

Table 5-431. PLLDIG_APLL_SW_DIS_DELAY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	cfg_plldig_auto_switch_delay	R/W	FA0h	Delay to switch the PLL clock source when the auto PLL switch mode is enable
15-0	cfg_apll_auto_switch_delay	R/W	FA0h	Delay to switch the PLL clock source when the auto PLL switch mode is enable

5.3.6.7 PLLDIG_APLL_SW_DIS_DELAY2 Register (Offset = 18h) [Reset = 00640064h]

PLLDIG_APLL_SW_DIS_DELAY2 is shown in Table 5-432.

Return to the Summary Table.

Table 5-432. PLLDIG_APLL_SW_DIS_DELAY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	cfg_apll_disable_delay	R/W	64h	Delay between the PLL clock source switching and disabling of the APLL
15-0	cfg_plldig_disable_delay	R/W	64h	Delay between the PLL clock source switching and disabling of the PLL DIG

5.3.6.8 PLLDIG_OVERRIDE Register (Offset = 1Ch) [Reset = X]

PLLDIG_OVERRIDE is shown in Table 5-433.

Return to the Summary Table.

Table 5-433. PLLDIG_OVERRIDE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	X	
5-3	cfg_ov_final_plldig_apll_mux_sel	R/W	0h	Override control for the fast clock src mux select 0x 0 = PLL DIG selected as fast clock 0x 7 = APLL selected as fast clock
2-0	cfg_sel_ov_final_plldig_apll_mux_sel	R/W	0h	Mux select control to select the override value of the fast clock src mux select 0x 0 = functional value selected for the fast clock src mux select 0x 7 = Override value selected for the fast clock src mux select

5.3.6.9 PLLDIG_STATUS Register (Offset = 20h) [Reset = X]

PLLDIG_STATUS is shown in Table 5-434.

Return to the Summary Table.

Table 5-434. PLLDIG_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	X	
8	plldig_lockmon	R	0h	PLL DIG lockmon status
7-2	RESERVED	R	X	
1-0	clkm_xtal_freq	R	0h	XTAL clock frequency status, 0x 0 = 25MHz 0x 1 = 40MHz 0x 2 = 26MHz 0x 3 = 38.4MHz

5.3.6.10 FAST_CLK_MUX_POSTDIV Register (Offset = 24h) [Reset = X]

FAST_CLK_MUX_POSTDIV is shown in Table 5-435.

Return to the Summary Table.

Table 5-435. FAST_CLK_MUX_POSTDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	X	
15-4	divr	R/W	0h	Divider value for FAST selected clock. Data should be loaded as multibit. For example: if divider value of 8 (1000) needs to be selected then '100010001000' should be configured to the register.
3-0	currdivr	R	1h	Status shows the current divider value choosen for FAST_CLK.

5.3.6.11 FAST_CLK_STATUS Register (Offset = 28h) [Reset = X]

FAST_CLK_STATUS is shown in Table 5-436.

Return to the Summary Table.

Table 5-436. FAST_CLK_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	X	
1-0	currclk	R	1h	Current Clock selected by GCM Clock Mux 0x 1 : PLLDIG 0x 2 : APLL

5.3.6.12 LOCK0_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0_KICK0 is shown in Table 5-437.

Return to the Summary Table.

- KICK0 component

Table 5-437. LOCK0_KICK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

5.3.6.13 LOCK0_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0_KICK1 is shown in Table 5-438.

Return to the Summary Table.

- KICK1 component

Table 5-438. LOCK0_KICK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

5.3.6.14 intr_raw_status Register (Offset = 1010h) [Reset = X]

intr_raw_status is shown in Table 5-439.

Return to the Summary Table.

Interrupt Raw Status/Set Register

Table 5-439. intr_raw_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

5.3.6.15 intr_enabled_status_clear Register (Offset = 1014h) [Reset = X]

intr_enabled_status_clear is shown in Table 5-440.

Return to the Summary Table.

Interrupt Enabled Status/Clear register

Table 5-440. intr_enabled_status_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

ADVANCE INFORMATION

5.3.6.16 intr_enable Register (Offset = 1018h) [Reset = X]

intr_enable is shown in Table 5-441.

Return to the Summary Table.

Interrupt Enable register

Table 5-441. intr_enable Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

5.3.6.17 intr_enable_clear Register (Offset = 101Ch) [Reset = X]

intr_enable_clear is shown in Table 5-442.

Return to the Summary Table.

Interrupt Enable Clear register

Table 5-442. intr_enable_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

Table 5-442. intr_enable_clear Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

5.3.6.18 eoi Register (Offset = 1020h) [Reset = X]

eoi is shown in Table 5-443.

Return to the Summary Table.

EOI register

Table 5-443. eoi Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

5.3.6.19 fault_address Register (Offset = 1024h) [Reset = 00000000h]

fault_address is shown in Table 5-444.

Return to the Summary Table.

Fault Address register

Table 5-444. fault_address Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

5.3.6.20 fault_type_status Register (Offset = 1028h) [Reset = X]

fault_type_status is shown in Table 5-445.

Return to the Summary Table.

Fault Type Status register

Table 5-445. fault_type_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	X	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_ 0000 = Supervisor read fault - priv = 1 dir = 1 dtype != 1 01_ 0000 = Supervisor write fault - priv = 1 dir = 0 00_ 1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_ 0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_ 0010 = User write fault - priv = 0 dir = 0 00_ 0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_ 0000 = No fault

5.3.6.21 fault_attr_status Register (Offset = 102Ch) [Reset = 00000000h]

fault_attr_status is shown in Table 5-446.

Return to the Summary Table.

Fault Attribute Status register

Table 5-446. fault_attr_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

5.3.6.22 fault_clear Register (Offset = 1030h) [Reset = X]

fault_clear is shown in Table 5-447.

Return to the Summary Table.

Fault Clear register

Table 5-447. fault_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	X	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

5.4 Memory Protection Unit (MPU)

The MPU performs memory protection checking for a CBA bus. It inputs a VBUSM or VBUSP bus, then checks the address against the fixed and programmable regions to see if the access is allowed. If allowed, the transfer is passed unmodified to the output VBUSM or VBUSP bus. If the transfer is illegal (fails the protection check), then the MPU does not pass the transfer to the output bus but rather services the transfer internally back to the input bus (to prevent a hang) returning the fault status to the requestor as well as generating an interrupt about the fault.

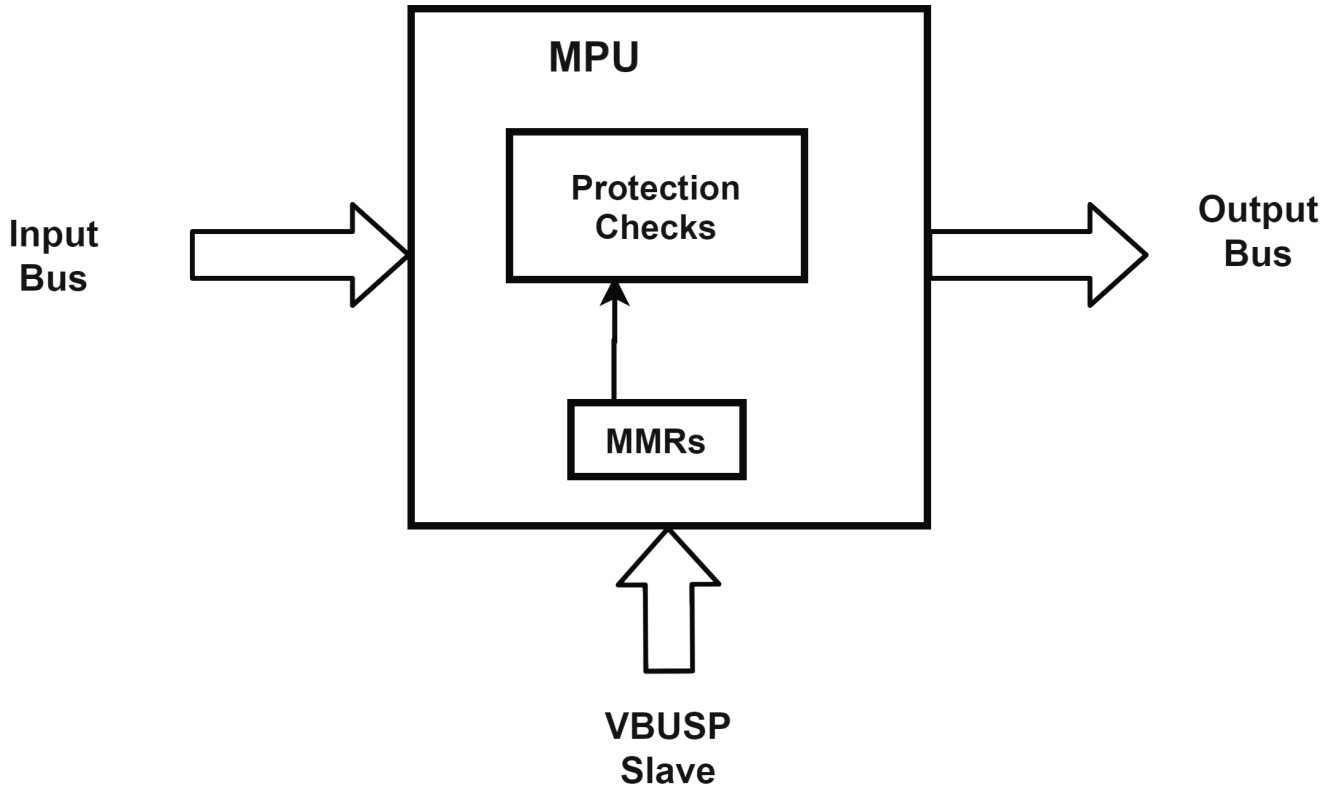


Figure 5-1361. MPU Top Level Diagram

The first check is the transfer's privID against the AID settings. There is an AID register bit for each possible privID (0 to 15) and an AIDX that covers privIDs not configured. The privID is used to lookup the associated AID bit. If the AID bit is 0, then the range does not cover that privilege level and the range is not checked (although other ranges with different AID setting will) for this transfer. If the AID bit is 1, then the range does cover that privilege level and the permissions are checked. The transfer secure and debug parameters are checked against the MPPA values to detect an allowed access. The two bits (NS and EMU) provide 3 permission levels. If the NS is set, the range is non-secure and any security or debug level may access the range. If the NS is not set, the range is secure only and only secure level accesses are allowed. In secure mode, if the EMU is set, debug accesses are allowed. If the EMU is not set, debug accesses are not allowed. For non-debug accesses, the read, write, and execute permissions are also checked. There is a set of permissions for supervisor mode and another for user mode. The "priv" attribute of the transfer determines which is checked. If priv = 1, the supervisor rwx bits are checked against the "dir" and "dtype" attributes of the transfer (read is dir = 1 and dtype not instruction, write is dir = 0, execute is dir = 1 and dtype = instruction). If priv = 0, the user rwx bits are checked against the same attributes. If the associated rwx bit for the type of transfer is 1, the transfer is allowed, but if the rwx bit is 0, the transfer is not allowed.

The function outputs whether the transfer is allowed or not. If the transfer address range (start to end address, or those with AID bits = 0) does not match any range, the transfer is either allowed or disallowed based on the configuration mode of the MPU to "assumed allowed" or "assumed disallowed" mode. If any of the overlapped ranges does not allow the access, the access is not allowed. Only when all the overlapped ranges allow the access is the access allowed. The final permissions are the lowest of each type of permission from any hit range (so if a transfer hits 2 ranges, one that is rw and one that is rx, then the final permission is just r).

The PrivID allocation is primarily a static allocation in the system.

Table 5-448. PrivID Grouping and Allocation

Master Group	PrivID Allocation
MSS Masters	

Table 5-448. PrivID Grouping and Allocation (continued)

Master Group	PrivID Allocation
MSS Cortex R5FA,B AXI-M (x2)	0x2
MSS eDMA TPTCs (x3)	0x2
Ethernet DMA	0x2
Debug Masters	
DebugSS JTAG	0x4
RS232	0x4
DMM	0x4
DSS Masters	
DSS MDMA Port	0x5
DSS eDMA TPTC (x4)	0x5
DSS eDMA TPTC (x2,x2,x2)	0x5
RSS Masters	
RSS ICSS-M Ports (x2)	0x6
RSS CSI2 A, B (x2)	0x6
RSS eDMA TPTCs (x2)	0x6

Figure 5-3. MPU Top Level Diagram

Memory Map (MPU)

The default registers are listed in Table 5-449. There can also be configured MMRs.

Table 5-449. Memory Map Registers

Address Offset	Register
0x000	Section 5.4.1
0x004	Section 5.4.2
0x010	Section 5.4.3
0x014	Section 5.4.4
0x018	Section 5.4.5
0x01C	Section 5.4.4.1
0x020	Section 5.4.6
0x024	Section 5.4.7
0x028 - 0x0FC	Reserved
0x100	Section 5.4.8
0x104	Section 5.4.9
0x108	Section 5.4.10
0x10C	Reserved
0x110 - 0x1FC	Reserved
0x200	Section 5.4.11
0x204	Section 5.4.12
0x208	Section 5.4.13
0x20C	Reserved
0x210 - 0x2FC	Additional Programmable Range MMRs
0x300	Section 5.4.14
0x304	Section 5.4.15

Table 5-449. Memory Map Registers (continued)

Address Offset	Register
0x308	Section 5.4.16

5.4.1 Revision Register (Base Address + 0x000)

The Revision Register contains the ID and revision information.

Table 5-450. Revision Register

Bits	Field	Type	Reset	Description
31:30	scheme	r/o	1	Scheme.
29:28	reserved	r/o	0	Always read as 0. Writes have no affect.
27:16	modID	r/o	0xe81	Module ID field.
15:11	revrtl	r/o	Any	RTL revision. Will vary depending on release.
10:8	revmaj	r/o	1	Major revision.
7:6	revcustom	r/o	0	Custom revision.
5:0	revmin	r/o	2	Minor revision.

5.4.2 Configuration Register (Base Address + 0x004)

The readonly Configuration register contains the configured values of the module.

Table 5-451. Configuration Register

Bits	Field	Type	Reset	Description
31:24	address_align	r/o	0	Address alignment for range checking. 0 = 1k 1 = 2k 2 = 4k …
23:20	num_fixed	r/o	0	Number of fixed address ranges.
19:16	num_prog	r/o	8	Number of programmable address ranges.
15:12	num_fixed_aids	r/o	0	Number of supported fixed AIDs. 0 = no specific fixed AIDs supported (all treated equally) N = PrivIDs from 0 to N-1 supported, others use AIDX
11:1	reserved	r/o	0	Always read as 0.
0	assumed_allowed	r/o	0	Assumed allowed mode. 0 = assumed disallowed 1 = assumed allowed

5.4.3 Interrupt Raw Status/Set Register (Base Address + 0x010)

The Interrupt Raw Status/Set register shows the interrupt status (before enabling) and allows setting of the interrupt status.

Table 5-452. Interrupt Raw Status/Set Register

Bits	Field	Type	Reset	Description
31:02:00	reserved	r/o	0	Always read as 0.
1	addr_err	w1ts	0	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	w1ts	0	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

5.4.4 Interrupt Enabled Status/Clear Register (Base Address + 0x014)

The Interrupt Enabled Status/Clear register shows the interrupt enabled status and allows clearing of the interrupt status.

Table 5-453. Interrupt Enabled Status/Clear Register

Bits	Field	Type	Reset	Description
31:2	reserved	r/o	0	Always read as 0.
1	enabled_ addr_err	w1tc	0	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_ prot_err	w1tc	0	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

5.4.4.1 Interrupt Enable Clear Register (Base Address + 0x01C)

The Interrupt Enable Clear register shows the interrupt enable and allows clearing of the interrupt enable. Reads return the Interrupt Enable register value, but writes can clear the interrupt enables.

Table 5-454. Interrupt Enable Clear Register

Bits	Field	Type	Reset	Description
31:02	reserved	r/o	0	Always read as 0.
1	addr_err_en_clr	w1tc	0	Addressing violation error enable. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_ en_clr	w1tc	0	Protection violation error enable. Write a 1 to clear the enable. Writing a 0 has no effect.

5.4.5 Interrupt Enable Register (Base Address + 0x018)

The Interrupt Enable register shows the interrupt enable value and allows setting the enable.

Table 5-455. Interrupt Enable Register

Bits	Field	Type	Reset	Description
31:02	reserved	r/o	0	Always read as 0.
1	addr_err_en	w1ts	0	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	w1ts	0	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

5.4.6 EOI Register (Base Address + 0x020)

The EOI register allows software to indicate when the end of interrupt service is complete. The eoi vector value is dependent on the interrupt handling.

Table 5-456. EOI Register

Bits	Field	Type	Reset	Description
31:8	reserved	r/o	0	Always read as 0.
7:0	eoi_vector	r/w	0	EOI vector value. Write this with the interrupt distribution value in the chip. This drives the mpu_eoi_vector output signal.

5.4.7 Interrupt Vector Register (Base Address + 0x024)

The Interrupt Vector register displays the interrupt vector returned from interrupt distribution.

Table 5-457. Interrupt Vector Register

Bits	Field	Type	Reset	Description
31:0	intr_vec	r/o	0	Interrupt vector. Reads mpu_intr_vector input signal.

5.4.8 Fixed Address Start Register (Base Address + 0x100)

The Fixed Address Start register holds the start address for the fixed range. The register is read only and may be invalid if the region mode is used to not perform an address compare. The size of the field is determined by the comp_width parameter.

5.4.9 Fixed Address End Register (Base Address + 0x104)

The Fixed Address End register holds the end address for the fixed range. The register is read only and may be invalid if the region mode is used to not perform an address compare. The width of the field is determined by the comp_width parameter.

5.4.10 Fixed MPPA Register (Base Address + 0x108)

The Fixed Address MPPA register holds the permissions for the fixed region. This register is writeable by a non-debug supervisor entity (priv = 1) only. If the NS bit is in secure mode (= 0), the register is also only writeable by a non-debug secure entity (secure = 1). The NS bit is only writeable by a secure entity. For debug accesses, the register is writeable only when NS = 1 or EMU = 1.

Table 5-458. Fixed MPPA Register

Bits	Field	Type	Reset	Description
31:26	reserved	r/o	0	Always read as 0.
25:10	AID15-0	r/w	input	AIDs checked for this region. Defaults to input value. 0 = AID is not checked for these permissions. 1 = AID is checked for these permissions.
9	AIDX	r/w	input	Additional AIDs checked. Defaults to input value.
8	reserved	r/o	0	Always read as 0.
7	ns	r/w	input	Non-secure permission. Defaults to input value.
6	emu	r/w	input	Debug permission. Defaults to input value.
5	sr	r/w	input	Supervisor read permission. Defaults to input value.
4	sw	r/w	input	Supervisor write permission. Defaults to input value.
3	sx	r/w	input	Supervisor executable permission. Defaults to input value.
2	ur	r/w	input	User read permission. Defaults to input value.
1	uw	r/w	input	User write permission. Defaults to input value.
0	ux	r/w	input	User executable permission. Defaults to input value.

5.4.11 Programmable Address Start Register N (Base Address + 0x200, 0x210, 0x220, …)

The Programmable Address Start register holds the start address for the range. This register is writeable by a supervisor entity (priv = 1) only. If the NS bit is in non-secure mode (= 0) in the associated MPPA register,

the register is also only writeable by a secure entity (secure = 1). The width of the field is determined by the comp_width parameter.

(Note: For GP device 'Bypass' is the default reset value and for HS-SE device 'Block' is the default reset value. For MPU_RSS_DSS2RSS and MPU_RSS_MSS2RSS the default reset value is 'Bypass' for all type of devices)

Refer to Default MPU Reset Configurations.

Table 5-459. Programmable Address Start Register

Bits	Field	Type	Reset	Description
31:N	start_addrN	r/w	input	Start address for range N. Defaults to input signal value.
N-1:0	reserved	r/o	0	Always read as 0.

5.4.12 Programmable Address End Register N (Base Address + 0x204, 0x214, 0x224, …)

The Programmable Address End register holds the end address for the range. This register is writeable by a supervisor entity (priv = 1) only. If the NS bit is in non-secure mode (= 0) in the associated MPPA register, the register is also only writeable by a secure entity (secure = 1). The field width is determined by the comp_width parameter.

(Note: For GP device 'Bypass' is the default reset value and for HS-SE device 'Block' is the default reset value. For MPU_RSS_DSS2RSS and MPU_RSS_MSS2RSS the default reset value is 'Bypass' for all type of devices)

Refer to Default MPU Reset Configurations.

Table 5-460. Programmable Address End Register

Bits	Field	Type	Reset	Description
31:N	end_addrN	r/w	input	End address for range N. Defaults to input signal value.
N-1:0	reserved	r/o	1s	Always read as all bits 1.

5.4.13 Programmable MPPA Register N (Base Address + 0x208, 0x218, 0x228, …)

The Programmable Address MPPA register holds the permissions for the region. This register is writeable by a non-debug supervisor entity (priv = 1) only. If the NS bit is in secure mode (= 0), the register is also only writeable by a non-debug secure entity (secure = 1). The NS bit is only writeable by a non-debug secure entity. For debug accesses, the register is writeable only when NS = 1 or EMU = 1.

(Note: For GP device 'Bypass' is the default reset value and for HS-SE device 'Block' is the default reset value. For MPU_RSS_DSS2RSS and MPU_RSS_MSS2RSS the default reset value is 'Bypass' for all type of devices)

Refer to Default MPU Reset Configurations.

Table 5-461. Programmable MPPA Register

Bits	Field	Type	Reset	Description
31:26	reserved	r/o	0	Always read as 0.
25:10	AID15-0	r/w	input	AIDs checked for this region. Defaults to input value. 0 = AID is not checked for these permissions. 1 = AID is checked for these permissions.
9	AIDX	r/w	input	Additional AIDs checked. Defaults to input value.
8	reserved	r/o	0	Always read as 0.
7	ns	r/w	input	Non-secure permission. Defaults to input value.
6	emu	r/w	input	Debug permission. Defaults to input value.
5	sr	r/w	input	Supervisor read permission. Defaults to input value.

Table 5-461. Programmable MPPA Register (continued)

Bits	Field	Type	Reset	Description
4	sw	r/w	input	Supervisor write permission. Defaults to input value.
3	sx	r/w	input	Supervisor executable permission. Defaults to input value.
2	ur	r/w	input	User read permission. Defaults to input value.
1	uw	r/w	input	User write permission. Defaults to input value.
0	ux	r/w	input	User executable permission. Defaults to input value.

5.4.14 Fault Address Register (Base Address + 0x300)

The Fault Address register holds the address of the first protection fault transfer.

Table 5-462. Fault Address Register

Bits	Field	Type	Reset	Description
31:0	fault_addr	r/o	0	Fault address.

5.4.15 Fault Status Register (Base Address + 0x304)

The Fault Status register holds the status and attributes of the first protection fault transfer.

Table 5-463. Fault Status Register

Bits	Field	Type	Reset	Description
31:24	id	r/o	0	Transfer ID.
23:16	mstid	r/o	0	controller ID.
15:13	reserved	r/o	0	Always read as 0.
12:09	privid	r/o	0	Privilege ID.
8	reserved	r/o	0	Always read as 0.
7	ns	r/o	0	Non-secure access.
6	reserved	r/o	0	Always read as 0.
5:00	fault_type	r/o	0	Fault type. 100000 = supervisor read fault 010000 = supervisor write fault 001000 = supervisor execute fault 000100 = user read fault 000010 = user write fault 000001 = user execute fault 111111 = relaxed cache linefill fault 010010 = relaxed cache writeback fault 000000 = no fault

5.4.16 Fault Clear Register (Base Address + 0x308)

The Fault Clear register allows the software to clear the current fault, so that another can be captured when this register is written.

Table 5-464. Fault Clear Register

Bits	Field	Type	Reset	Description
31:1	reserved	r/o	0	Always read as 0.
0	fault_clr	w/o	0	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

5.5 Reset

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

5.5.1 Reset Types and Sources

Table 5-465. Reset Types

Reset Type	Reset Source	Description
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 20 usec to reset the device.
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_PRCM:RST_SOFT_RESET register or watch dog module can generate this reset; additionally, the external pin WARM_RESET can also be used to trigger this reset. 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.

As listed in Table 5-465, various reset sources can generate the different resets used inside the device to reset various components and submodules.

5.5.2 Reset Domains

The device can be divided into various reset domains. 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 5-466.

Table 5-466. Reset Domains

Reset Domains	Description	Resets
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\u0002on 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
Application subsystem reset domain	This reset subdomain controls the reset to the application susbsystem and the modules inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset
Front-End Controller subsystem reset domain	This reset subdomain controls the reset to the front-end controller susbsystem and the modules inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset

Table 5-466. Reset Domains (continued)

Reset Domains	Description	Resets
Hardware Accelerator subsystem reset domain	This reset subdomain controls the reset to the HWA susbsystem and the modules inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset

5.5.3 Reset Cause Registers

The reset cause is maintained in three places:

- TOP_PRCM - device level reset cause register
- FEC_SS_RCM - FECSS level reset cause register
- APP_SS_RCM - APPSS level reset cause register

This register (xWRLx432:TOP_PRCM:SYS_RST_CAUSE) maintains the reset cause at the device level.

Table 5-467. TOP_PRCM Reset Cause Register

wWRLx432:TOP_PRCM:SYS_RST_CAUSE	Filed Name	Description
Bit#16	SYS_RST_CAUSE_SYS_RST_CAUSE_CLR	Clear's the sys_rst_cause register 0x0 -> sys_rst_cause capture enable 0x1 -> sys_rst_cause reg clear and disable
Bit#<2:0>	SYS_RST_CAUSE_SYS_RST_CAUSE	System Reset Cause register 3'b001 - POR reset 3'b010 - Warm reset due to soft register 3'b100 - Warm reset due to wdog

Note

On a STC_POR reset, the SYS_RST_CAUSE_SYS_RST_CAUSE will be 0x0. This field will only be set to 3'b001 on the true POR reset.

This register (xWRLx432:APP_RCM:RST_CAUSE) maintains the reset cause at the APPSS level.

Table 5-468. APP_RCM Reset Cause Register

Bits	Field Name	Description
7:0	RST_CAUSE_COMMON	Reset cause register for APP CPU • 000_0000 - All cleared • 000_0001 - Power On Reset (PoR) • 000_0010 - Subsystem Reset (Combination of Warm Reset initiated from PRCM using xWRLx432:TOP_PRCM:RST_APP_PD_SOFT_RESET and PoR reset) • 000_0100 - STC RESET • 000_1000 - Reserved • 001_0000 - CPU Only Reset triggered by writing to xWRLx432:APP_RCM:RST_FSM_TRIG< `RST_FSM_TRIG_CPU` > • 010_0000 - Core Reset initiated from PRCM using xWRLx432:TOP_PRCM:RST_SOFT_APP_CORE_SYSRESET_REQ (reset CPU unconditionally - by debugger) or xWRLx432:TOP_PRCM:APP_CORE_SYSRESET_PARAM_WAKEUP_OUT_STATE • 100_0000 - Reserved

This is cleared using - **xWRLx432:APP_RCM:RST_CAUSE_CLR**

ADVANCE INFORMATION

6 Processors and Accelerators

xWRLx432 includes the following processor core and accelerators:

- Application Subsystem Cortex M4
- Front End Controller Radar Subsystem Cortex M3
- Radar Processing Hardware Accelerators

6.1 Applications Subsystem Cortex M4

Main ARM SubSystem Features

The APPSS supports the following features:

- ARM Cortex M4 RISC Processor (Single core)
  - ARMv7-M architecture profile.
  - Support for NVIC (Nested Vectored Interrupt Controller) to achieve low interrupt latency
  - Ability to execute code from internal memory (via I-code/D-code) or external memories.
- ARM Cortex M4 based subsystem, primarily responsible for
  - Secure boot
  - Firewall controls
  - Device bootup sequence including clock configuration
  - Security controls
- Read-only Memory (ROM) to allow boot sequence, authentication and provide security service
  - 96 KB Size
  - Root of trust Secure boot operation
  - Includes a 48KB secure ROM – used only in HS device, Firewalled otherwise in GP device.
  - Accessible only to Cortex-M4
- Local memory bank for Runtime operation
  - 512 kb of dedicated RAM and 256 kb of shared RAM.
  - Used to store both code and data.
  - External world can access these memories if allowed via Firewall.
- Local 32-bit VBUSP Interconnect
  - Low latency access from Cortex-M4 to ROM/RAM.
  - Firewall policies enforced at the external host target port to restrict access.
  - External 32-bits VBUSP controller and target interface
  - Ability for the Cortex-M4 to reach any memory within the SoC
- SoC VBUSP Interconnect
  - Controls and Configures the firewalls within the SoC
- Interrupts
  - Support for upto 64 interrupts
- Timer Module
  - 2 instances of DMTimer module
  - 32-bit up counter
  - 64-bits can be achieved by cascading two timer modules
- WatchDog Timer
  - RTI module to perform windowed WDT operation
  - Can issue warm reset to the SoC
  - Can issue a reset to the HSM subsystem
- Debug
  - DAP based debug interface to ARM core
  - Full debug support inside ARM core. All debug functionality is present including data matching for watchpoint generation.
  - Limited Support for external ARM cross trigger interface (CTI) by providing interface signal to halt.
- Secure DMA
  - Dedicated EDMA with 1 TPCC, 2 TPTC
  - Accessible only to Cortex M4

6.1.1 APPSS Memory Organization

Figure 6-1 shows the memory organization for the CPU subsystems in APPSS. Total 512KB of dedicated RAM is provided to Cortex M4F core split across 3 target ports – 1x 256KB and 2x 128KB banks- as shown in Figure 6-1.

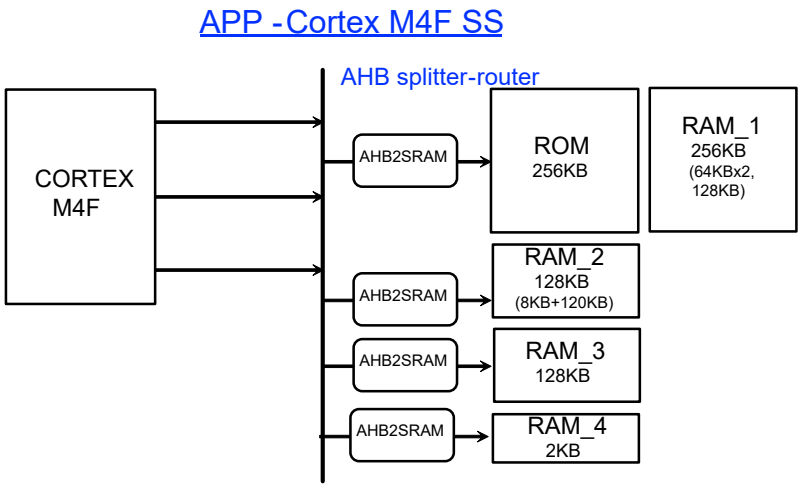


Figure 6-1. APPSS Memory Map

6.2 Radar Processing Hardware Accelerator

The xWRLx432 device incorporates Radar Hardware Accelerators HWA1.2 to offload the pre-processing computations.

Radar accelerator details are mainly covered in Chapter "Radar Hardware Accelerator 1.2", which consists of 3 parts.

6.2.1 Key Features

The main features of the Radar Hardware Accelerator are as follows:

1. Fast FFT computation, with programmable FFT sizes (powers of 2) up to 1024-pt complex FFT
2. Internal FFT bit width of 24 bits (for each I and Q) for good Signal to Quantization noise ratio (SQNR) performance, with fully programmable butterfly scaling at every radix-2 stage for user flexibility
3. Built-in capabilities for simple pre-FFT processing – specifically, programmable windowing, basic interference zeroing-out, and basic BPM removal
4. Magnitude (absolute value) and log-magnitude computation capability
5. Flexible data flow and data sample arrangement to support efficient multidimensional FFT operations and transpose accesses as required
6. Chaining and looping mechanism to sequence a set of accelerator operations one-after-another with minimal intervention from the main processor
7. DC Subtraction with user programmed values or with computed values using DC Estimation
8. Interference zero-out with localization using user programmed thresholds or computed thresholds with Interference statistics
9. CFAR-CA detector support (linear and logarithmic). CFAR-OS detector support (Logarithmic)
10. Compression/Decompression using block floating-point and EGE
11. Miscellaneous other capabilities of the accelerator:
    a. Stitching two or four 1024-point FFTs to get the equivalent of 2048-point or 4096-point FFT for industrial level sensing applications where large FFT sizes are required
    b. Slow DFT mode, with resolution equivalent to 16K size FFT, for FFT peak interpolation purposes (for example, range interpolation)
    c. Complex vector multiplication and Dot product capability for vectors up to 512 in size

6.3 Front-end Controller Subsystem Cortex M3

Cortex M3 core's prime responsibility is to sequence the different tasks requested by the application core and handle all the design dependent configuration of RF and Analog FE modules which application core and Customer's System application engineer need not know. Key tasks handled by the Cortex M3 core are Initialization of RF/ANA modules, Calibration, PRCM requests from host and safety monitoring which are done through DFP APIs.

6.3.1 FECSS Memory Organization

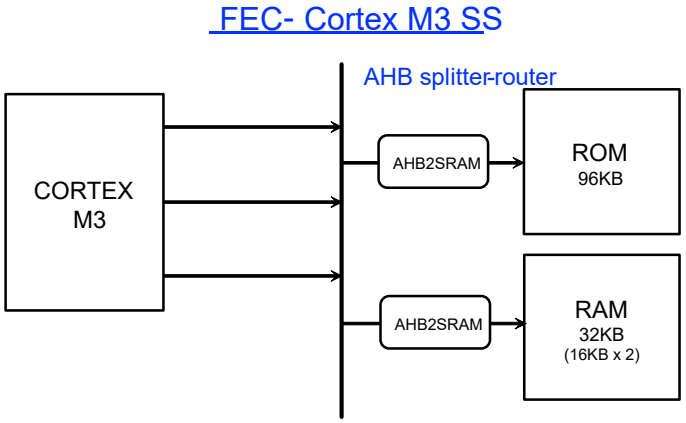


Figure 6-2. FECSS Memory Map

6.4 RF and Analog Subsystem

The RF and analog subsystem includes the 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 TX can be operated simultaneously for beam forming in BPM mode or individually in TDM mode. Similarly, the device allows configuring the number of receive channels based on application and power requirements. For system power saving, RF and analog subsystems can be put into low power mode configuration.

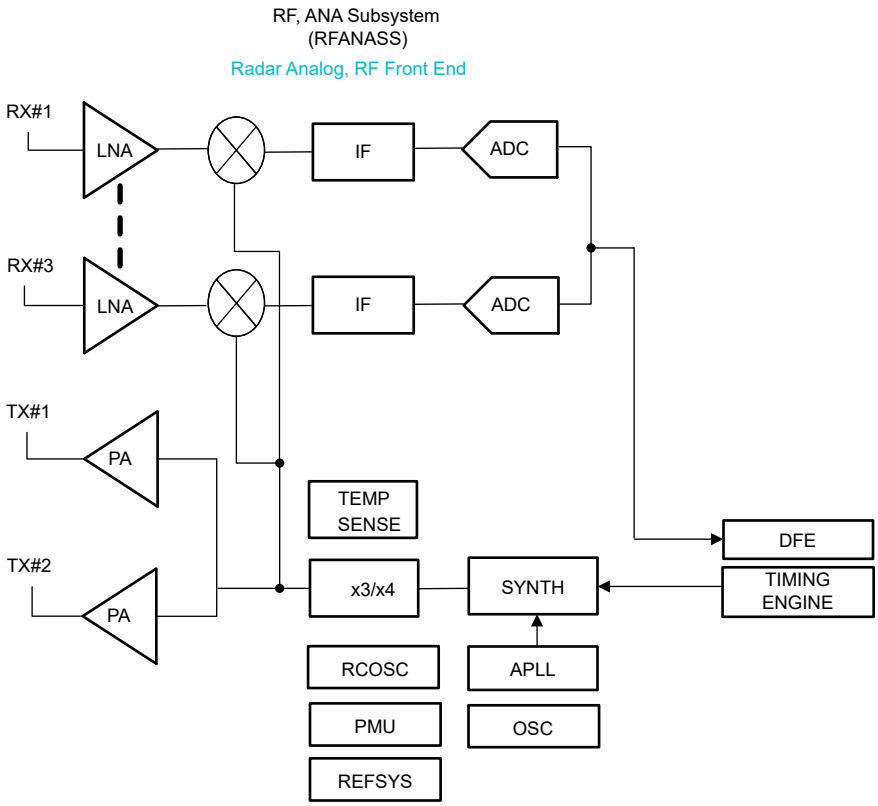


Figure 6-3.

6.4.1 Clock Subsystem

The xWRL6432 clock subsystem generates 57 to 63.9GHz from an input reference from a crystal. It has a built-in oscillator circuit followed by a clean-up PLL and a RF synthesizer circuit. The output of the RF synthesizer is then processed by an X3 multiplier to create the required frequency in the 57 to 63.9 spectrum. The RF synthesizer output is modulated by the timing engine block to create the required waveforms for effective sensor operation.

The clean-up PLL also provides a reference clock for the host processor after system wakeup.

The clock subsystem also has built-in mechanisms for detecting the presence of a crystal and monitoring the quality of the generated clock.

Figure 6-4 describes the clock subsystem.

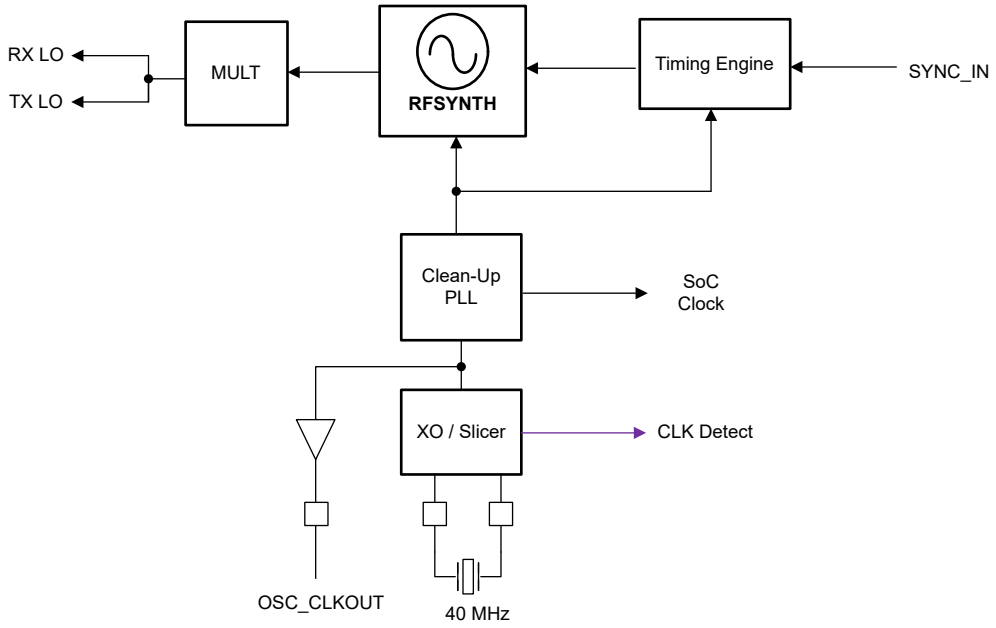


Figure 6-4. Clock Subsystem

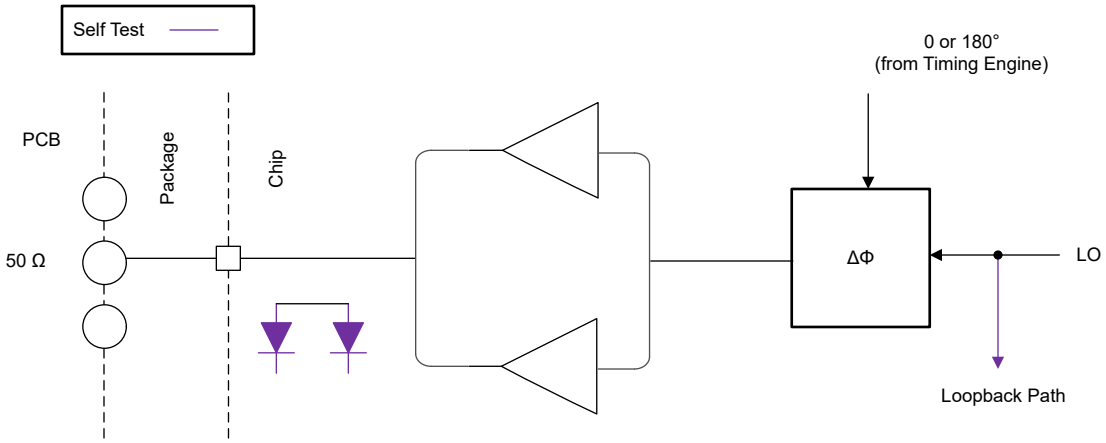
6.4.2 Transmit Subsystem

The xWRL6432 transmit subsystem consists of two parallel transmit chains, each with independent phase and amplitude control. The device supports binary phase modulation for MIMO radar.

The transmit chains also support programmable backoff for system optimization.

Figure 6-5 describes the transmit subsystem.

Figure 6-5. Transmit Subsystem (Per Channel)



6.4.3 Receive Subsystem

The xWRLx432 receive subsystem consists of three parallel channels. A single receive channel consists of an LNA, mixer, IF filtering, ADC conversion, and decimation. All three receive channels can either operate simultaneously OR can be powered down individually based on system power needs and application design.

The xWRLx432 device supports a real baseband architecture, which uses real mixer, single IF and ADC chains to provide output for each receiver channel. The device is targeted for fast chirp systems. The band-pass IF chain has configurable lower cutoff frequencies above 175 kHz and can support bandwidths up to 5MHz.

Figure 6-6 describes the receive subsystem.

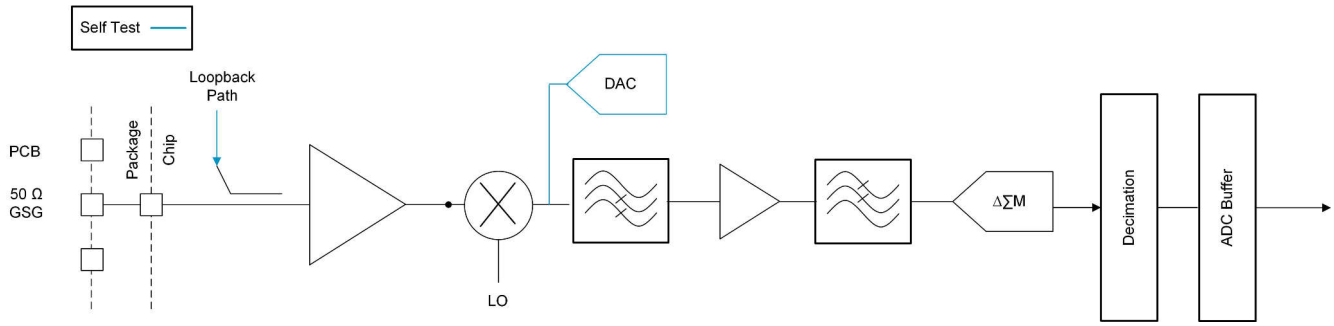


Figure 6-6. Receive Subsystem (Per Channel)

ADVANCE INFORMATION

7 Interrupts

This section lists the various interrupts sources supported in the different subsystem of xWRLx432 device.

7.1 APPSS Non-Maskable Interrupt

Non-maskable Interrupt (NMI)	ESM_HI_IRQ

7.2 APPSS Interrupt Map

Interrupt Num	Interrupt Source	Description	PULSE/LEVEL
0	WIC_IRQ	Interrupt line same as WIC_LINE[3]	PULSE
1	APPSS_ESM_LO_IRQ	APPSS ESM Low Priority Interrupt	LEVEL
2	FEC_INTR0	SW Interrupts to APPSS Cortex M4F (IPC Resp)	PULSE
3	FEC_INTR1	SW Interrupts to APPSS Cortex M4F (IPC ACK)	PULSE
4	FEC_INTR2	SW Interrupts from FEC SS	PULSE
5	FEC_INTR3	SW Interrupts from FEC SS	PULSE
6	APPSS_GIO_INT0	High Level Interrupt. Aggregated interrupt. Assumption is that only one relevant pin would be enabled to generate this interrupt.	LEVEL
7	APPSS_GIO_INT1	Low Level Interrupt. Aggregated interrupt. Assumption is that only one relevant pin would be enabled to generate this interrupt.	LEVEL
8	APPSS_SCI1_INT0	16 sources internally-mapped within the IP to either of two intr lines	LEVEL
9	APPSS_SCI1_INT1	SCI1_INT1: 16 sources internally- mapped within the IP to either of two intr linesMCAN_WAKEUP_INTR : Interrupt asserted when MCAN receives a dominant bit and wakeup is enabled in IP. IP does not support wakeup interruptThe mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL[23]	LEVEL
10	APPSS_LIN_INT0	LIN_INT0: 16 sources internally- mapped within the IP to either of two intr linesMCAN_WAKEUP_INTR : Interrupt asserted when MCAN receives a dominant bit and wakeup is enabled in IP. IP does not support wakeup interrupt The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL[24]	LEVEL
11	APPSS_LIN_INT1	16 sources internally- mapped within the IP to either of two intr lines	LEVEL
12	MUXED_APPSS_DCC_DONE_INT _AND_BIST_INTR	The following are multiplexed to this interrupt line: APPSS_DCC_DONE, CM3_LBIST_INTR, CM4_LBIST_INTR, PBIST_INTR, using mux sel APPSS_CTRL::DCC_DONE_AND_BIST_MUX_ SEL. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	PULSE
13	APPSS_QSPI_INTR_REQ	APPSS QSPI Interrupt	LEVEL
14	APPSS_SPI_IRQ_REQ	APPSS SPI Interrupt	LEVEL
15	APPSS_TPCC1_INTAGG	APPSS_TPCC1 Aggregated Functional Interrupt	LEVEL
16	MUXED_APPSS_TPCC1_ERRAGG_CTI _TRIGOUT2	The following are multiplexed to this interrupt line: APPSS_TPCC1_ERRAGG (APPSS_TPCC1 Aggregated Error Interrupt) and CTI_TRIGOUT2The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
17	APPSS_TPCC2_INTAGG	APPSS_TPCC1 Aggregated Functional Interrupt	LEVEL
18	MUXED_APPSS_TPCC2_ERRAGG_CTI _TRIGOUT3	The following are multiplexed to this interrupt line: APPSS_TPCC2_ERRAGG (APPSS_TPCC2 Aggregated Error Interrupt) and CTI_TRIGOUT3The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL

Interrupt Num	Interrupt Source	Description	PULSE/LEVEL
19	MUXED_APPSS_I2C_INT_AND_CM3_LBIST_INTR _AND_CM4_LBIST_INTR_AND_ PBIST_INTR	The following are multiplexed to this interrupt line: APPSS_I2C_INT, CM3_LBIST_INTR, CM4_LBIST_INTR, PBIST_INTR. using mux sel APPSS_CTRL::I2C_INT_AND_BIST_MUX_SEL. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	PULSE
20	APPSS_MCRC_INT	APPSS MCRC Interrupt	LEVEL
21	APPSS_MCAN_INT0	MCANA first interrupt	LEVEL
22	APPSS_MCAN_INT1	MCANA second interrupt	LEVEL
23	APPSS_MCAN_FE_INT1	MCANA message filter interrupt1	PULSE
24	APPSS_MCAN_FE_INT2	MCANA message filter interrupt2	PULSE
25	APPSS_MCAN_FE_INT3	MCANA message filter interrupt3	PULSE
26	APPSS_MCAN_FE_INT4	MCANA message filter interrupt4	PULSE
27	APPSS_MCAN_FE_INT5	MCANA message filter interrupt5	PULSE
28	MUXED_APPSS_MCAN_FE_INT6 _AND_SPI2_IRQ_REQ	The following are multiplexed to this interrupt line: MCAN_FE_INT6 and SPI2_IRQ_REQ. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SELMCAN_WAKEUP_INTR : Interrupt asserted when MCAN receives a dominant bit and wakeup is enabled in IP. IP does not support wakeup interrupt The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL[25] and APP_CTRL:APPSS_IRQ_REQ_SEL[16]IRQ_REQ_SEL[25[, IREQ_REQ_SEL[16]0 , 0 => MCAN_FE_INT60.1 => SPI2_IRQ_REQ1, x => MCAN_WAKEUP_INTR	PULSE/LEVEL
29	MUXED_APPSS_MCAN_FE_INT7 _AND_DEBUGSS_TXDATA_AVAIL	The following are multiplexed to this interrupt line: DEBUGSS_TX_DATA_AVAIL and MCAN_FE_INT7. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	PULSE
30	MUXED_FECSS_CHIRPTIMER_CHIRP_START _AND_CHIRP_END	The following are multiplexed to this interrupt line: Chirp timer chirp_start, Chirptimer chirp_end CHIRP_END. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	PULSE
31	MUXED_FECSS_CHIRPTIMER_BURST_START _AND_BURST_END	The following are multiplexed to this interrupt line: Chirp timer burst_start, Chirp timer burst_end inputs and BURST_END. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	PULSE
32	FECSS_CHIRPTIMER_FRAME_END	Frame End Interrupt from Frame Timer	PULSE
33	FECSS_FRAMETIMER_FRAME_START	Frame Start Interrupt from Frame Timer	PULSE
34	MUXED_FECSS_CHIRP_AVAIL_IRQ _AND_ADC_VALID_START_AND_SYNC_IN	The following are multiplexed to this interrupt line: Chirp avail IRQ, Chirptimer ADC_VALID_START and SYNC_IN IO . The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	PULSE
35	FECSS_FRAME_START_OFFSET_INTR_TIME1	FRAME_START_OFFSET_INTR_TIME1 from Frame timer.	PULSE
36	FECSS_FRAME_START_OFFSET_INTR_TIME2	FRAME_START_OFFSET_INTR_TIME2 from Frame timer	PULSE
37	FECSS_FRAME_START_OFFSET_INTR_TIME3	FRAME_START_OFFSET_INTR_TIME3 from Frame timer	PULSE
38	FECSS_BURST_START_OFFSET_TIME	BURST_START_OFFSET_TIME from Chirp timer	PULSE
39	SW_IRQ0	APPSS SW Interrupts	PULSE
40	SW_IRQ1	APPSS SW Interrupts	PULSE
41	SW_IRQ2	APPSS SW Interrupts	PULSE
42	SW_IRQ3	APPSS SW Interrupts	PULSE

Interrupt Num	Interrupt Source	Description	PULSE/LEVEL
43	MUXED_APPSS_RTI1_RTI2_INT_REQ0	The following are multiplexed to this interrupt line: int_req0 from RTI1 and RTI2(WD) modules. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
44	MUXED_APPSS_RTI1_RTI2_INT_REQ1	Multiplexed int_req1 from RTI1 and RTI2(WD) modules. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
45	MUXED_APPSS_RTI1_RTI2_INT_REQ2	The following are multiplexed to this interrupt line: int_req2 from RTI1 and RTI2(WD) modules. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
46	MUXED_APPSS_RTI1_RTI2_INT_REQ3	The following are multiplexed to this interrupt line: int_req3 from RTI1 and RTI2(WD) modules. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
47	MUXED_APPSS_RTI1_RTI2_TBINT _AND_GPADC_IFM_DONE	The following are multiplexed to this interrupt line: tb_int from RTI1 and RTI2(WD) modules and gpadc interrupt. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
48	MUXED_APPSS_RTI1_CAP_EVT0 _AND_RTI2_CAP_EVT0_AND_PWM_INT 0	The following are multiplexed to this interrupt line: between CAP_EVT0 from RTI1, CAP_EVT0 from RTI2 and PWM_INT[0]. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
49	MUXED_APPSS_RTI1_CAP_EVT1 _AND_RTI2_CAP_EVT1_AND_PWM_INT 1	The following are multiplexed to this interrupt line: between CAP_EVT1 from RTI1, CAP_EVT1 from RTI2 and PWM_INT[1]. The mux sel is APP_CTRL:APPSS_IRQ_REQ_SEL	LEVEL
50	HWASS_LOOP_INT	HWA Loop Complete Interrupt	PULSE
51	HWASS_PARAMDONE_INT	HWA Param Done Interrupt	PULSE
52	SHA_S_INT	Reserved	
53	SHA_P_INT	Reserved	
54	TRNG_INT	Reserved	
55	PKAE_INT	Reserved	
56	AES_S_INT	Reserved	
57	AES_P_INT	Reserved	
58	HSM_MB_INT0	Reserved	
59	HSM_MB_INT1	Reserved	
60	APPSS_MUXED_PERIPH_ACCESS_ER RAGG	MPU aggregated errors	LEVEL
61	APPSS_WKUP_INTR	from PRCM (IO/Wkup timer). connected to Cortex M4F's WIC input also.	PULSE
62	APPSS_SCI2_INT0	UART2 (SCI2) Interrupt#0	LEVEL
63	APPSS_SCI2_INT1	UART2 (SCI2) Interrupt#1	LEVEL

7.3 APPSS ESM Interrupt Map

ESM Group 1

ESM GROUP1	Define Name	Description
31	ACESS_ERROR	Aggregated address access error.
30	TOPSS_AGG_ERR	Aggregated TOPSS Errors- FRC undefined state error
29	PLL_DIG_LOC_MON	Lock monitor signal from the PLL_DIG
28	PM_AGG_ERR	Aggregated Error Signals from PM. RCOSC10M_GOOD , RCOSC32K_GOOD, SUPPLY_OK etc

ESM GROUP1	Define Name	Description
27	ANA_AGG_ERR	Aggregated Error Signals from Analog- Saturation detection, APLL Lock Monitor
26	EFUSE_ERR	EFUSE Error
25	QSPI_WR_ERR	QSPI write error
24	MPU_PROT_AGG_ERR	Aggregated MPU Protection Error for MPUs
23	APPSS_MCRC_ERR	MCRC Comparision Error
22	APPSS_DCC_ERR	DCC frequency comparision error
21	APPSS_MCAN_AGG_ERR	Aggregated MCAN Errors- MCAN_SERR - Single Bit correctable error indication for MCANA Message Memory- MCAN_UERR- Multi Bit uncorrectable error indication for MCANA Message Memory- MCAN_TS_ERR - MCANA Timestamping Error
20	APPSS_TPCC_AGG_ERR	APPSS_TPCC Aggregated Error Interrupt- TPCC Error- TPTC Error for all TPTCs connected to TPCC- Read and Write Config Space Access error to TPCC- Read and Write Config Space Access error or all TPTCs connected to TPCC
19	HWASS_TPCC_AGG_ERR	HWASS_TPCC Aggregated Error Interrupt- TPCC Error- TPTC Error for all TPTCs connected to TPCC- Read and Write Config Space Access error to TPCC- Read and Write Config Space Access error or all TPTCs connected to TPCC
18	CM3_LBIST_ERR	LBIST Error indication for Cortex M3
17	CM4F_LBIST_ERR	LBIST Error indication for Cortex M4F
16	HWA_AGG_ERR	Aggregated HWA Errors- iping/ipong/ oping/opong memories access error- FSM Lock step error- window ram parity error- iping/ipong/oping/opong memories parity error- shared memory invalid address access error
15	HWA_LOCAL_RAM_ECC_AGG_SERR	Aggregated ECC multi-bit error from HWA local RAMs- HWA param ram- ADCBUF ping/pong memories- HWASS TPTC1 FIFO- HWASS TPTC2 FIFO
14	HWA_LOCAL_RAM_ECC_AGG_UERR	Aggregated ECC single-bit error from HWA local RAMs- HWA param ram- ADCBUF ping/pong memories- HWASS_TPTC1 FIFO- HWASS_TPTC2 FIFO
13	HWASS_SHARED_RAM_ECC_AGG_SERR	ECC single-bit error indication from shared memory banks owned by HWASS
12	HWASS_SHARED_RAM_ECC_AGG_UERR	ECC multi-bit (uncorrectable) error indication from shared memory banks owned by HWASS
11	FECSS_AGG_ERR	Aggregated Error from DFE, Timing Engines and other FECSS modules- Parity error from timing engine- undefined_st_entered error from timing engine
10	FECSS_ECC_AGG_SERR	Aggregated ECC single-bit error (from other FECSS RAMs than CPU RAMs)- GPADC ram
9	FECSS_ECC_AGG_UERR	Aggregated ECC multi-bit error (from other FECSS RAMs than CPU RAMs)-GPADC ram

ESM GROUP1	Define Name	Description
8	FECSS_CM3_RAM_ECC_AGG_SERR	Aggregated ECC single-bit error from FECSS CPU RAMs and 96 KB shared ram when it is shared
7	FECSS_CM3_RAM_ECC_AGG_UERR	Aggregated ECC multi-bit error from FECSS CPU RAMs and 96 KB shared ram when it is shared
6	APPSS_ECC_AGG_SERR	Aggregated ECC single-bit error from APPSS RAMs (other than CPU RAMs)- APPSS_TPTC1 FIFO- APPSS_TPTC2 FIFO
5	APPSS_ECC_AGG_UERR	Aggregated ECC multi-bit (Uncorrectbale) error from APPSS RAMs (other than CPU RAMs)- APPSS_TPTC1 FIFO- APPSS_TPTC2 FIFO
4	APPSS_CM4F_RAM_ECC_AGG_SERR	Aggregated ECC single-bit error from APPSS Cortex M4F ROM, RAMs and 128KB/256 KB shared ram when it is shared
3	APPSS_CM4F_RAM_ECC_AGG_UERR	Aggregated ECC multi-bit (Uncorrectable) error from APPSS Cortex M4F ROM, RAMs and 128KB/256 KB shared ram when it is shared
2	APPSS_WDT_NMI	APPSS Watch dog timer non maskable irq
1	HSM_ESM_LO	RESERVED for future scalability
0	HSM_ESM_HI	RESERVED for future scalability

ESM Group 2

Not supported.

ESM Group 3

ESM GROUP3	Define Name	Description
7 to 4	RESERVED	RESERVED
3	CLKM_LIMP_MODE	from PRCM. Error for Reference clock not ticking.
2	APPSS_CM4F_RAM_ECC_AGG_UERR	Aggregated ECC multi-bit (Uncorrectable) error from APPSS Cortex M4F ROM, RAMs and 128KB/256 KB shared ram when it is shared
1	EFUSE_AUTOLOAD_ERR	Efuse autoload error
0	RESERVED	RESERVED

8 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, or by the hardware FFT accelerator for the post processing.

For the application software, the ADC buffer (either ping or pong) is seen as a single memory at the base address.

RadarSS generates Chirp Parameter (CP) and Chirp Quality (CQ) data along with ADC data. Interface Control Document (ICD) explains CP/CQ data format and steps to enable this feature. Refer ICD for the latest supported features towards CP/CQ/ADC of the device.

8.1 Functional Description

Figure 8-1 shows the block diagram of the ADC buffer scheme. The two data input sources to the ADC buffer are:

- Raw ADC output data from the digital front end (DFE)
- Ramp pattern data from the test pattern generator

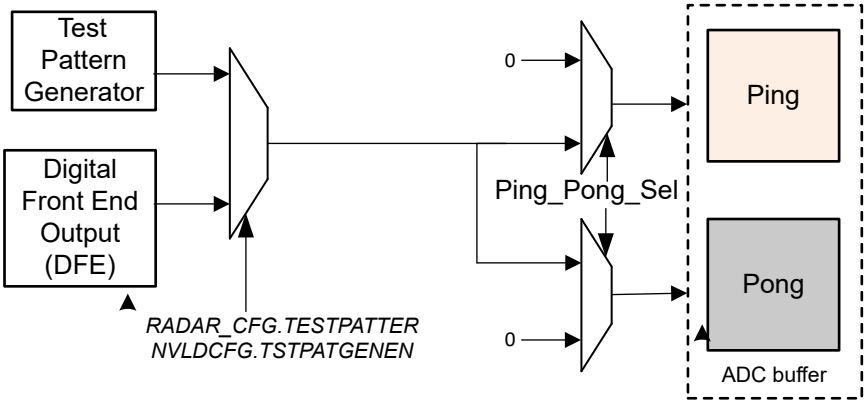


Figure 8-1. ADC Buffer Block Diagram

8.1.1 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 APP_HWA_ADCBUF_CTRL address space:

1. Single-chirp mode
2. Multi-chirp mode
3. Continuous mode

The DFE data from the three 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 8-2. ADC buffer control logic generates the Ping_Pong_Sel signal, as shown in Figure 8-2, which controls whether the data is written into either ping or pong buffer. Data write can start from either the ping or pong buffer.

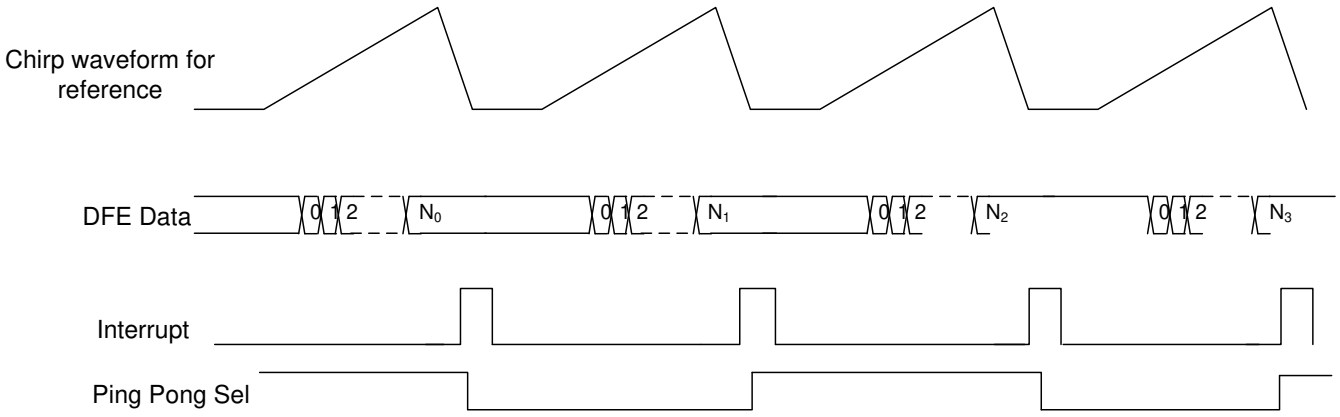


Figure 8-2. Single-Chirp Mode

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 and the pong buffer are configured in the register field ADCBUFNUMCHRPPING.

ADC Buffer Single-Chirp and Multi-Chirp Mode Programming Sequence 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 8-1. 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
Configure the number of samples to be stored in each ping/pong buffer as N	ADCBUFCFG4.ADCBUFNUMCHRPP1NG

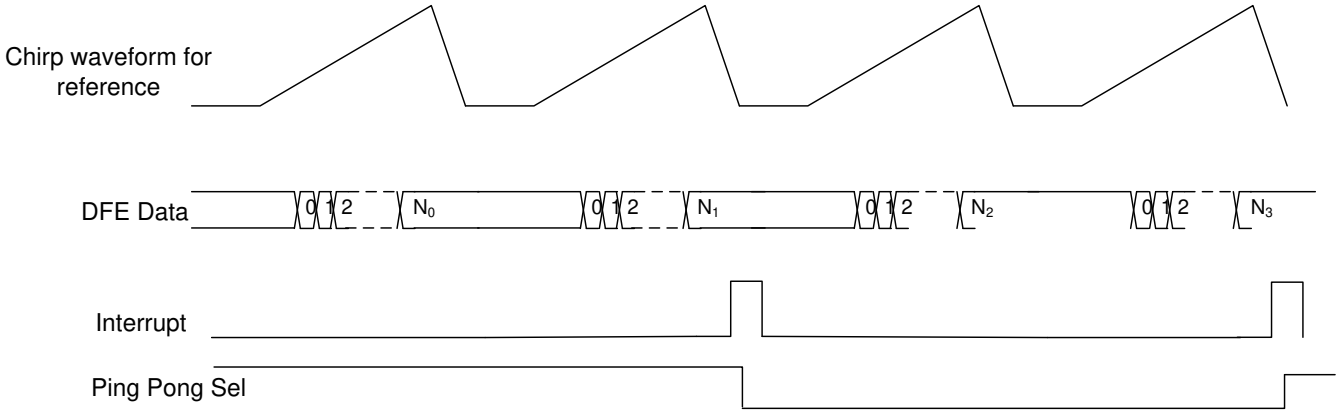


Figure 8-3. Multi-Chirp Mode

In continuous mode, where the FMCW transceiver is configured to output a single frequency tone in the range of X-Y GHz (where X is start and Y is end frequency supported by the device), '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 APP_HWA_ADCBUF_CTRL.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.

ADC Buffer Continuous Mode Programming Sequence shows the programming sequence for ADC buffer continuous mode.

Table 8-2. 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
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

8.1.2 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 8-1. 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 RADAR_CFG 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 RSS_CTRL address space and test pattern generator-related registers for further information.

8.1.3 ADC Buffer Data Formats

The data is written in the following formats to the ADC buffer:

- Non-interleaved data format

8.1.3.1 Non-Interleaved Data Format

In non-interleaved mode storage, each channel data is stored in different memory locations, as shown in Table 8-3.

Table 8-3. 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)

ADVANCE INFORMATION

9 Quad Serial Peripheral Interface (QSPI)

9.1 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 controller only.

The one QSPI in the device is primarily intended for fast booting from quad-SPI flash memories. Figure 9-1 shows the QSPI module overview.

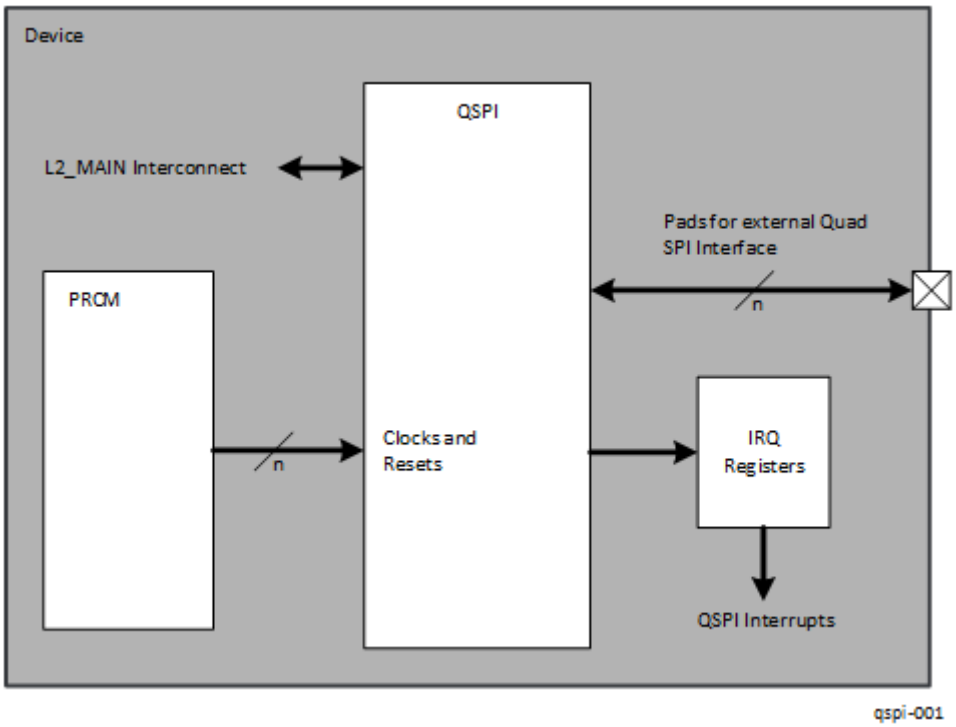


Figure 9-1. QSPI Overview

The QSPI supports the following features:

- General SPI features:
  - Programmable clock divider
  - Max four pin interface
  - Programmable length (from 1 to 128 bits) of the words transferred
  - Programmable number (from 1 to 4096) of the words transferred
  - 1 external chip-select signal
  - Support for 1 pin Write. Dual or quad writes are not supported
  - Support for 1-, 2-, or 4-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 L2_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
  - External flash support of up to 8 MB
  - 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 for memory mapped registers used to configure QSPI controller and not SPI content accesses)
- 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.

9.2 QSPI Environment

Figure 9-2 shows a typical connection of the QSPI module to the external quad-SPI flash memory.

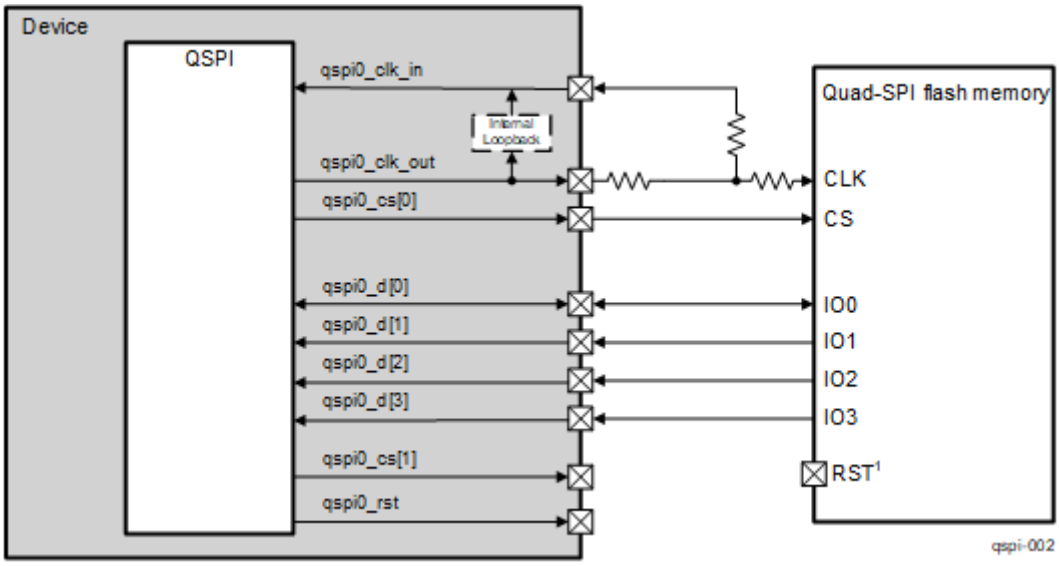


Figure 9-2. QSPI Connected to an External Quad-SPI Flash Memory

1. External flash memories that are larger than 128 Mb require an external reset pin for correct operation after SoC PORz reset. This reset must be triggered upon board reset to ensure the flash is in the correct state upon boot.

Table 9-1 lists and describes the QSPI I/O signals.

Table 9-1. QSPI I/O Signals

QSPI Signal/Pad name	I/O[1]	Description					
		3-pin[2] SPI Read (Single Read)	3-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Single Read)	4-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Dual Read)	6-pin[2] SPI Read (Quad Read)
qspi0_d[0]	IO	Used as SPI data input	Used as SPI data output	Not used	Used as SPI data output	Used as SPI data input 0	Used as SPI data input 0
qspi0_d[1]	I	Not used	Not used	Used as SPI data input	Not used	Used as SPI data input 1	Used as SPI data input 1
qspi0_d[2]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 2
qspi0_d[3]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 3
qspi0_sclk	O	Clock for the external SPI device					
qspi0_cs[0]	O	External SPI device chip-select 0					

Table 9-1. QSPI I/O Signals (continued)

QSPI Signal/Pad name	I/O[1]	Description					
		3-pin[2] SPI Read (Single Read)	3-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Single Read)	4-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Dual Read)	6-pin[2] SPI Read (Quad Read)
qspi0_rtclk	I	The qspi0_sclk output must be connected to the qspi0_rtclk input, and is used for controlling the timing of the read return data when the QSPI module operates in Mode 0. In case Mode 3 is used, there is no need to connect the qspi0_sclk to the qspi0_rtclk.					

(1) I = Input; O = Output
(2) This is the pin count at the SPI flash memory side. The pin count at the device side is increased by one because of the qspi0_rtclk signal. References to the pin count throughout this chapter consider the pin count at the SPI flash memory side.

Note

To ensure proper timing, precise layout and routing requirements must be followed. For layout and routing requirements for all QSPI signals, see section “PCB Guidelines” of the device Data Manual.

9.3 QSPI Functional Description

9.3.1 QSPI Block Diagram

Initial 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 qspi0_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 9-3 shows the QSPI block diagram.

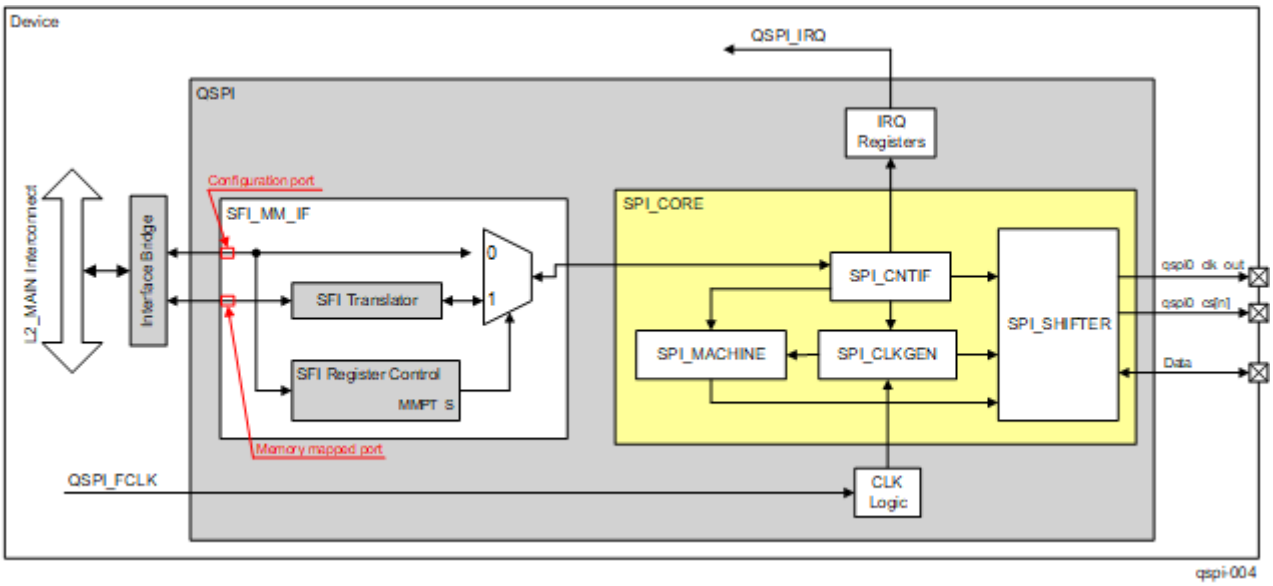


Figure 9-3. QSPI Block Diagram

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:

ADVANCE INFORMATION

- 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 L2_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 externally 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.

9.3.1.1 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 9-3) to connect directly to the SPI_CORE block, or allows the memory-mapped port (also shown in Figure 9-3) 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.

9.3.1.2 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.

Note

The input to the SFI Memory Mapped Protocol Translator is 23 address lines. Therefore, the SFI mode of operation supports external flash size of up to 8MB

9.3.1.3 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 (qspi0_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

ADVANCE 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 qspi0_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 qspi0_d[1] is sent to qspi0_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 qspi0_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 qspi0_d[0] pad to be used as an output.

9.3.1.4 SPI Clock Generator

The SPI clock generator uses the QSPI_FCLK clock as an input, and generates the qspi0_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 qspi0_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 9-4 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.

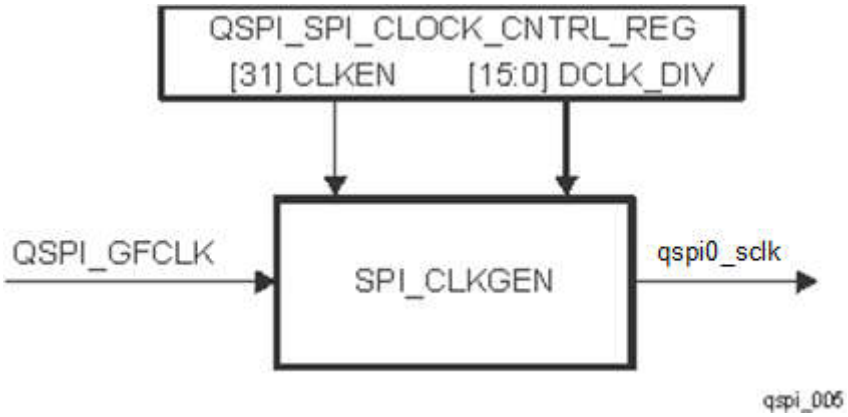


Figure 9-4. SPI_CLKGEN Block

9.3.1.5 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 qspi0_cs[n] (n = 0 to 1) 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 increments the value of the WDCNT bit field from 0x0 and starts shifting data in or out again. This is repeated until the WDCNT bit field reaches the frame length (QSPI_SPI_CMD_REG[11:0] FLEN), 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 qspi0_cs[n] (n = 0 to 1) 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 qspi0_cs[n] (n = 0 to 1) becomes inactive when all words are shifted or when the frame terminates earlier.

9.3.1.6 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 qspi0_sclk clock depending on the SPI clock mode selected. Table 9-2 lists the four defined clock modes of operation for the QSPI.

Table 9-2. 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 qspi0_sclk clock. Data output generated on falling edge of qspi0_sclk clock
1	0	1	Data input captured on rising edge of qspi0_sclk clock. Data output generated on rising edge of qspi0_sclk clock
2	1	0	Data input captured on rising edge of qspi0_sclk clock. Data output generated on rising edge of qspi0_sclk clock

Table 9-2. SPI Clock Modes Definition (continued)

Mode	Settings in the QSPI_SPI_DC_REG Register		Description
	Value of the CKP bits	Value of the CKPH bits	
3	1	1	Data input captured on falling edge of qspi0_sclk clock. Data output generated on falling edge of qspi0_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 9-5 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 qspi0_sclk clock cycles after the corresponding qspi0_cs[n] (n = 0 to 1) 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.

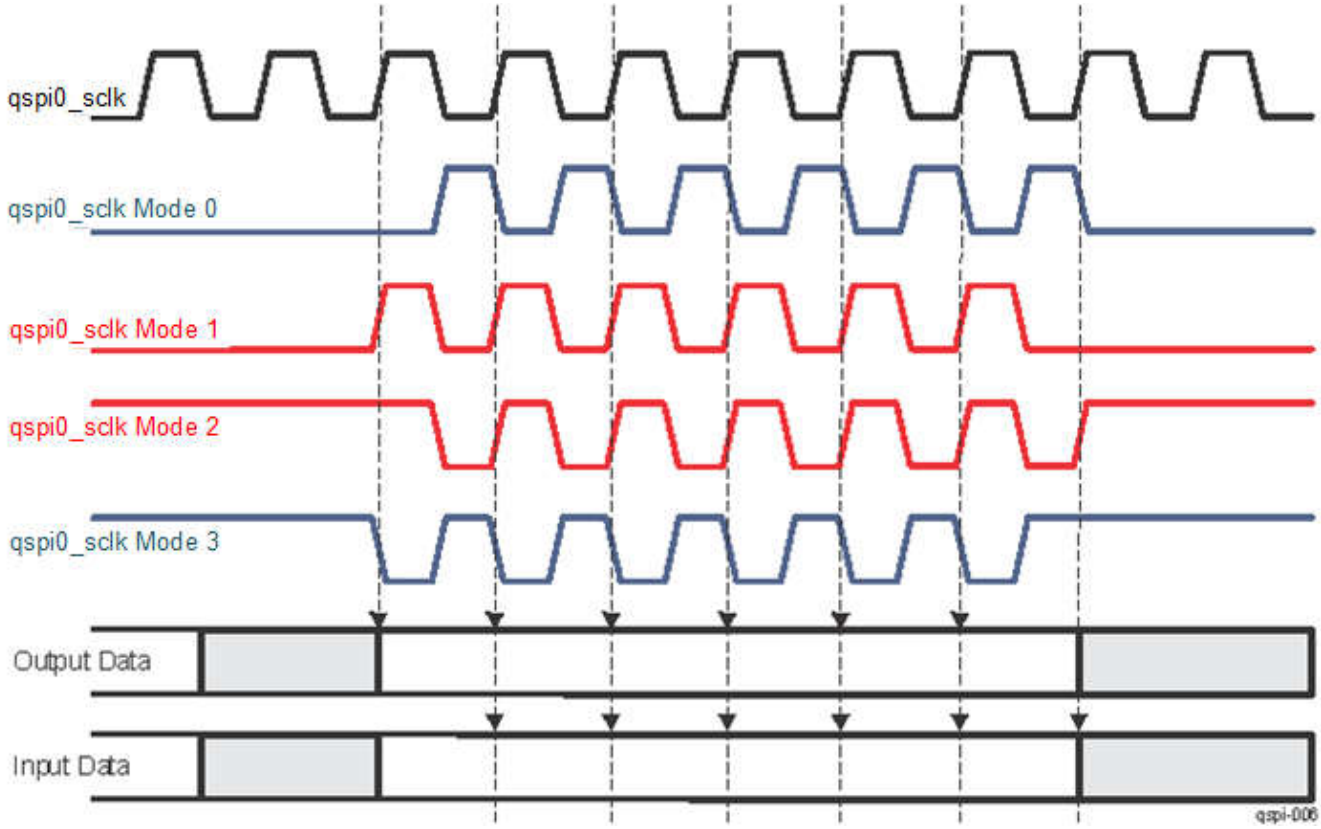


Figure 9-5. SPI Clock Modes

9.3.2 QSPI Clock Configuration

The QSPI complies with the PRCM peripheral-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.

9.3.3 QSPI Interrupt Requests

Figure 9-6 shows a logical representation of the QSPI interrupt generation scheme.

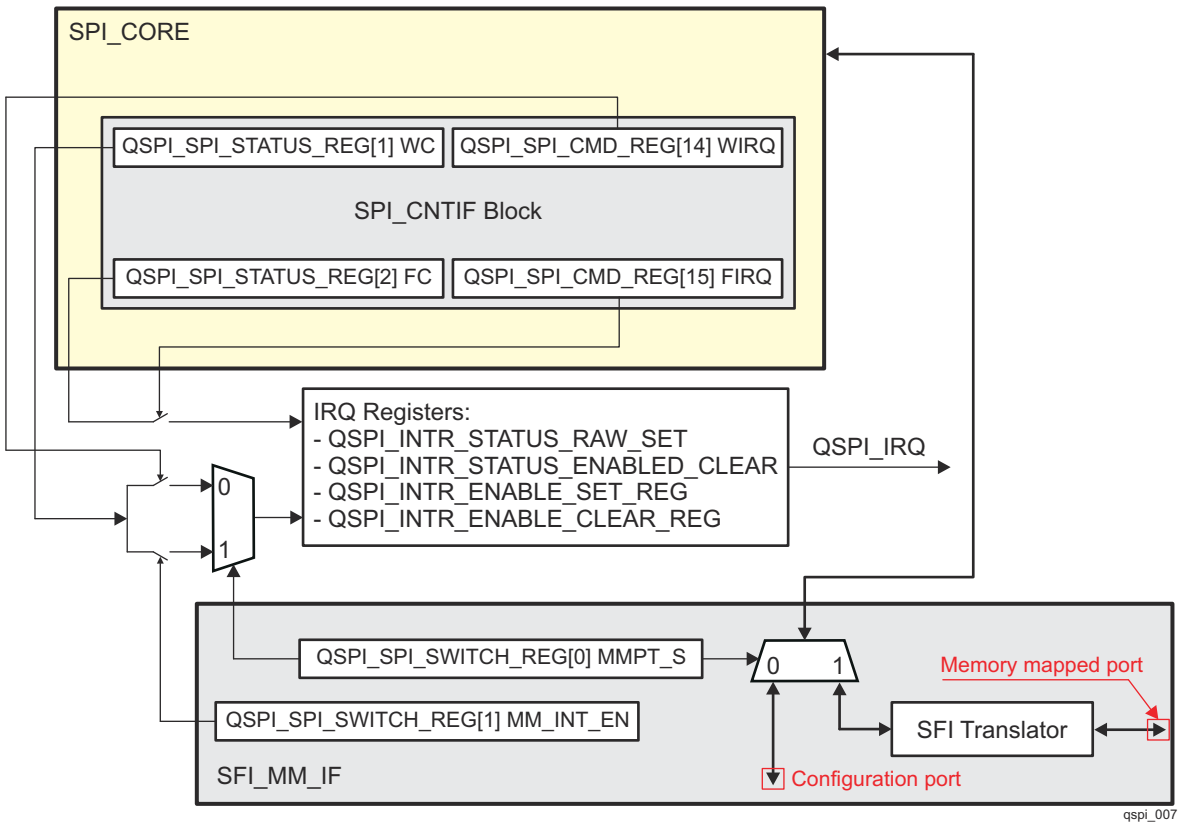


Figure 9-6. Logical Representation of the QSPI Interrupt Generation Scheme

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 9-3 lists the event flags and the corresponding mask bits of the sources which can cause interrupts.

Table 9-3. QSPI Events

Event Flag	Event Mask	Description
QSPI_INTR_STATUS_RAW_SET[1] WIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[1] WIRQ_ENA QSPI_SPI_STATUS_REG[1] WC	QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[1] WIRQ_ENA_CLR QSPI_SPI_CMD_REG[14] WIRQ	Word complete interrupt event. Asserted each time after a word is transferred or received.
QSPI_INTR_STATUS_RAW_SET[0] FIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[0] FIRQ_ENA QSPI_SPI_STATUS_REG[2] FC	QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[0] FIRQ_ENA_CLR QSPI_SPI_CMD_REG[15] FIRQ	Frame complete interrupt event. Asserted each time after a frame is transferred or received.

Note

QSPI_IRQ can also be used to trigger DMA events

9.3.4 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 supported external SPI devices. The L2_MAIN start address at which the configuration port is available is 0x4820 0000. The second memory region is associated mainly with the memory-mapped port and is used for communication directly with one of the two supported external SPI devices. The memory region for device 1 starts at 0x6000 0000 and the memory region for device 2 starts at 0x6200 0000

It is important to keep in mind that the configuration port provides an access to all the QSPI registers listed in the register summary. 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.

9.4 QSPI Registers

9.4.1 QSPI Register Summary

Table 9-4. QSPI Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset
QSPI_PID	R	32	0x0000 0000
QSPI_SYSCONFIG	RW	32	0x0000 0010
QSPI_INTR_STATUS_RAW_SET	RW	32	0x0000 0020
QSPI_INTR_STATUS_ENABLED_CLEAR	RW	32	0x0000 0024
QSPI_INTR_ENABLE_SET_REG	RW	32	0x0000 0028
QSPI_INTR_ENABLE_CLEAR_REG	RW	32	0x0000 002C
QSPI_INTC_EOI_REG	RW	32	0x0000 0030
QSPI_SPI_CLOCK_CNTRL_REG	RW	32	0x0000 0040
QSPI_SPI_DC_REG	RW	32	0x0000 0044
QSPI_SPI_CMD_REG	RW	32	0x0000 0048
QSPI_SPI_STATUS_REG	R	32	0x0000 004C
QSPI_SPI_DATA_REG	RW	32	0x0000 0050
QSPI_SPI_SETUP0_REG	RW	32	0x0000 0054
QSPI_SPI_SETUP1_REG	RW	32	0x0000 0058
QSPI_SPI_SETUP2_REG	RW	32	0x0000 005C
QSPI_SPI_SETUP3_REG	RW	32	0x0000 0060
QSPI_SPI_SWITCH_REG	RW	32	0x0000 0064
QSPI_SPI_DATA_REG_1	RW	32	0x0000 0068
QSPI_SPI_DATA_REG_2	RW	32	0x0000 006C
QSPI_SPI_DATA_REG_3	RW	32	0x0000 0070

9.4.2 QSPI Register Description

Table 9-5. QSPI_PID

Address Offset	0x0000 0000
Description	Revision register
Type	R

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

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI Internal data

Table 9-6. QSPI_SYSCONFIG

Address Offset	0x0000 0010
Description	
Type	RW

31	30	29	28	27	26	25	24	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																												IDLE_MODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x2

Bits	Field Name	Description	Type	Reset
3:2	IDLE_MODE	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 0x1: No-idle mode 0x2: Smart-idle mode 0x3: Reserved.	RW	0x2
1:0	RESERVED		R	0x0

Table 9-7. QSPI_INTR_STATUS_RAW_SET

Address Offset	0x0000 0020
Description	This register contains raw interrupt status flags.
Type	RW

31	30	29	28	27	26	25	24	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																														WIRQ_RAW	FIRQ_RAW

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		RW	0x0
1	WIRQ_RAW	Word Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0
0	FIRQ_RAW	Frame Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0

Table 9-8. QSPI_INTR_STATUS_ENABLED_CLEAR

Address Offset	0x0000 0024
Description	This register contains status flags of the enabled interrupts.
Type	RW

31	30	29	28	27	26	25	24	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																														WIRQ_ENA	FIRQ_ENA

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	WIRQ_ENA	Word Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the word interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0
0	FIRQ_ENA	Frame Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the frame interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0

Table 9-9. QSPI_INTR_ENABLE_SET_REG

Address Offset	0x0000 0028
Description	This register enables the interrupts.
Type	RW

31	30	29	28	27	26	25	24	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																														WIRQ_ENA_SET	FIRQ_ENA_SET

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_SET	Word interrupt enable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt enabled Write: 0x0: Has no effect 0x1: Enables the word interrupt	RW	0x0
0	FIRQ_ENA_SET	Frame interrupt enable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Enables the frame interrupt	RW	0x0

Table 9-10. QSPI_INTR_ENABLE_CLEAR_REG

Address Offset	0x0000 002C
Description	This register disables the interrupts.

Table 9-10. QSPI_INTR_ENABLE_CLEAR_REG (continued)

Type	RW

31	30	29	28	27	26	25	24	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																														WIRQ_ENA_CLR	FIRQ_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_CLR	Word interrupt disable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the word interrupt	RW	0x0
0	FIRQ_ENA_CLR	Frame interrupt disable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the frame interrupt	RW	0x0

Table 9-11. QSPI_INTC_EOI_REG

Address Offset	0x0000 0030
Description	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if a new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).
Type	RW

31	30	29	28	27	26	25	24	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																															

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the interrupt outputs. There is one interrupt output. Write 0x0 after servicing the interrupt to be able to generate another interrupt if pulse interrupts are used. Any other write value is ignored.	RW	0x0

Table 9-12. QSPI_SPI_CLOCK_CNTRL_REG

Address Offset	0x0000 0040
Description	This register controls the external SPI clock generation. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit. RW

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

CLKEN	RESERVED	DCLK_DIV

Bits	Field Name	Description	Type	Reset
31	CLKEN	External SPI clock (qspi1_sclk) enable. 0x0: The qspi1_sclk clock is turned off 0x1: The qspi1_sclk clock is enabled	RW	0x0
30:16	RESERVED		R	0x0
15:0	DCLK_DIV	Divide ratio for the external SPI clock (qspi1_sclk)	RW	0x0

Table 9-13. QSPI_SPI_DC_REG

Address Offset	0x0000 0044
Description	This register controls the different modes for each output chip select. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	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																											DD0			CKPH0	CSP0	CKP0

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:3	DD0	Data delay for chip select 0 0x0: Data is output on the same cycle as the qspi1_cs[0] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[0] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[0] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[0] goes active	RW	0x0
2	CKPH0	Clock phase for chip select 0. If CKP0 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP0 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
1	CSP0	Chip select polarity for chip select 0. 0x0: Active low 0x1: Active high	RW	0x0
0	CKP0	Clock polarity for chip select 0. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0

Table 9-14. QSPI_SPI_CMD_REG

Address Offset	0x0000 0048
Description	This register sets up the SPI command. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	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		CSNUM		RESERVED		WLEN							CMD			FIRQ	WIRQ	RESERVED		FLEN											

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	CSNUM	Device select. Sets the active chip select for the current transfer. 0x0: Chip Select 0 active 0x1: Chip Select 1 active 0x2: Chip Select 2 active 0x3: Chip Select 3 active	RW	0x0
27:26	RESERVED		R	0x0
25:19	WLEN	Word length. Sets the size of the individual transfers from 1 to 128 bits. When a word length greater than 32 bits is configured, not only the QSPI_SPI_DATA_REG register, but also the QSPI_SPI_DATA_REG_1, QSPI_SPI_DATA_REG_2, QSPI_SPI_DATA_REG_3 are used. One or all of these registers are used depending on the length of words transferred. 0x0: 1 bit 0x1: 2 bits ... 0x7F: 128 bits	RW	0x0
18:16	CMD	Transfer command. 0x0: Reserved 0x1: 4-pin Read Single 0x2: 4-pin Write Single 0x3: 4-pin Read Dual 0x4: Reserved 0x5: 3-pin Read Single 0x6: 3-pin Write Single 0x7: 6-pin Read Quad	RW	0x0
15	FIRQ	Frame complete interrupt enable. 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0
14	WIRQ	Word complete interrupt enable 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0
13:12	RESERVED		R	0x0
11:0	FLEN	Frame Length. 0x0: 1 word 0x1: 2 words ... 0xFFF: 4096 words	RW	0x0

Table 9-15. QSPI_SPI_STATUS_REG

Address Offset	0x0000 004C
Description	This register contains indicators to allow the user to monitor the progression of a frame transfer. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	R

31	30	29	28	27	26	25	24	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								WDCNT								RESERVED													FC	WC	BUSY

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WDCNT	Word count. This field will reflect the 1-4096 words transferred	R	0x0
15:3	RESERVED		R	0x0
2	FC	Frame complete. This bit is set after the transmision of all the requested words completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Transfer is not complete 0x1: Transfer is complete	R	0x0
1	WC	Word complete. This bit is set after each word transfer completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Word transfer is not complete 0x1: Word transfer is complete	R	0x0
0	BUSY	Busy bit. Active transfer in progress. This bit is only set during an active word transfer. Between words it is cleared. 0x0: Idle 0x1: Busy	R	0x0

Table 9-16. QSPI_SPI_DATA_REG

Address Offset	0x0000 0050
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the first 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	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																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 9-17. QSPI_SPI_SETUP0_REG

Address Offset	0x0000 0054
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 0 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
Type	RW

31	30	29	28	27	26	25	24	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			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2

Bits	Field Name	Description	Type	Reset
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 9-18. QSPI_SPI_SETUP1_REG

Address Offset 0x0000 0058

Description This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 1 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.

Type RW

31	30	29	28	27	26	25	24	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			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0

Bits	Field Name	Description	Type	Reset
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 9-19. QSPI_SPI_SETUP2_REG

Address Offset	0x0000 005C
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 2 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
Type	RW

31	30	29	28	27	26	25	24	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				NUM_D_BITS				WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 9-20. QSPI_SPI_SETUP3_REG

Address Offset	0x0000 0060
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 3 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
Type	RW

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

ADVANCE INFORMATION

RESERVED	NUM_D_BITS	WCMD	RESERVED	READ_TYPE	NUM_D_BYTES	NUM_A_BYTES	RCMD

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 9-21. QSPI_SPI_SWITCH_REG

Address Offset	0x0000 0064
Description	This register allows initiators to switch control of the SPI core port between the configuration port and the SFI translator. In addition, an interrupt enable field is defined which is used to enable or disable word complete interrupt generation in memory mapped mode.
Type	RW

31	30	29	28	27	26	25	24	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																														MM_INT_EN	MMPT_S

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	MM_INT_EN	Memory mapped mode interrupt enable. 0x0: Word complete interrupt is disabled during memory mapped operations 0x1: Word complete interrupt is enabled for memory mapped operations	RW	0x0
0	MMPT_S	MPT select. 0x0: Configuration port is selected to control the SPI_CORE. 0x1: SFI translator is selected to control the SPI_CORE.	RW	0x0

Table 9-22. QSPI_SPI_DATA_REG_1

Address Offset	0x0000 0068
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the second 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	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																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 9-23. QSPI_SPI_DATA_REG_2

Address Offset	0x0000 006C
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the third 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	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																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 9-24. QSPI_SPI_DATA_REG_3

Address Offset	0x0000 0070
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the fourth 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
Type	RW

31	30	29	28	27	26	25	24	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																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

10 Radar Hardware Accelerator

The *RadarHardware Accelerator User's Guide* (in three parts) describes the Radar Hardware Accelerator architecture, features, and operation of various blocks and their register descriptions. The purpose is to enable the user to understand the capabilities offered by the Radar Hardware Accelerator and to program it appropriately to achieve the desired functionality.

This user's guide is divided into three parts. The first part (this document) provides an overview of the overall architecture and features available in the Radar Hardware Accelerator. The main features, such as, windowing, FFT, and log-magnitude are covered in this part.

The second part of the user's guide covers additional features like CFAR-CA, CFAR-OS and other advanced usage possibilities. The second part of the user's guide is optional and can be skipped if the user is interested only in the FFT computation capability.

The third part covers the compression and decompression engines.

10.1 RadarHardware Accelerator - Part 1

Part 1 of the user's guide is organized as follows: Section 1 covers the introduction and high-level architecture. Section 2, Section 3, and Section 4 describe the state machine, trigger mechanisms, input/output formatting, and general framework for using the accelerator. Section 5 describes the primary computational unit features, namely, windowing, FFT, and log-magnitude.

10.1.1 RadarHardware Accelerator

This section provides an overview of the Radar Hardware Accelerator 1.2. The section covers the key features of the accelerator and overall architecture.

10.1.1.1 Introduction

The Radar Hardware Accelerator 1.2 is a hardware IP that enables off-loading the burden of certain frequently used computations in FMCW radar signal processing from the main processor. It is well known that 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 with minimal intervention from main processor.

10.1.1.2 Key Features

The main features of the Radar Hardware Accelerator are as follows:

1. Fast FFT computation, with programmable FFT sizes (powers of 2) up to 1024-pt complex FFT
2. Internal FFT bit width of 24 bits (for each I and Q) for good Signal to Quantization noise ratio (SQNR) performance, with fully programmable butterfly scaling at every radix-2 stage for user flexibility
3. Built-in capabilities for simple pre-FFT processing – specifically, programmable windowing, basic interference zeroing-out, and basic BPM removal
4. Magnitude (absolute value) and log-magnitude computation capability
5. Flexible data flow and data sample arrangement to support efficient multidimensional FFT operations and transpose accesses as required
6. Chaining and looping mechanism to sequence a set of accelerator operations one-after-another with minimal intervention from the main processor
7. DC Subtraction with user programmed values or with computed values using DC Estimation
8. Interference zero-out with localization using user programmed thresholds or computed thresholds with Interference statistics
9. CFAR-CA detector support (linear and logarithmic). CFAR-OS detector support (Logarithmic)
10. Compression/Decompression using block floating-point and EGE
11. Miscellaneous other capabilities of the accelerator:
    a. Stitching two or four 1024-point FFTs to get the equivalent of 2048-point or 4096-point FFT for industrial level sensing applications where large FFT sizes are required
    b. Slow DFT mode, with resolution equivalent to 16K size FFT, for FFT peak interpolation purposes (for example, range interpolation)
    c. Complex vector multiplication and Dot product capability for vectors up to 512 in size

10.1.1.3 High-level Architecture

The Radar Hardware Accelerator module is loosely coupled to the main processor (ARM® Cortex®-M4 in the mmWave xWRLx432 device). The accelerator is connected to a 64-bit bus that is present in the main processor system, as shown in Figure 10-1

The Radar Hardware Accelerator module comprises an accelerator engine and four memories, each of 16KB size, which are used to send input data to and pull output data from the accelerator engine. These memories are referred to as *localmemories* of the Radar Accelerator (ACCEL_MEM). For convenience, these four local memories are referred to as ACCEL_MEM0, ACCEL_MEM1, ACCEL_MEM2, and ACCEL_MEM3.

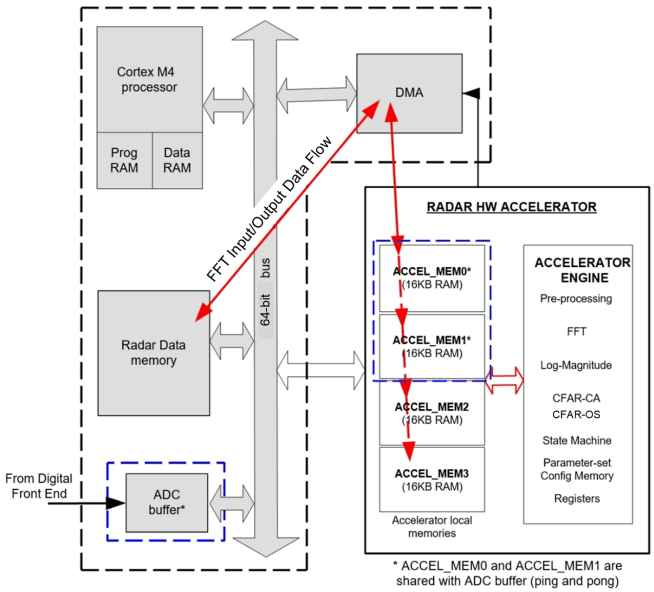


Figure 10-1. RadarHardware Accelerator 1.2 (xWRLx432 Device)

10.1.1.3.1 High-level Data flow

The typical data flow is that the DMA module is used to bring samples (for example, FFT input samples) into the local memories of the Radar Hardware Accelerator, so that the main accelerator engine can access and process these samples. Once the accelerator processing is done, the DMA module reads the output samples from the local memories of the Radar Hardware Accelerator and stores them back in the Radar data memory for further processing by the main processor. In Figure 10-1, the red arrow shows data movement from the Radar data memory into the accelerator local memories for the FFT and other processing steps. The red arrow also shows the output samples from the accelerator being picked up by the DMA and written back into the Radar data memory for further processing by the main processor.

Note that in the mmWave xWRLx432 device, the Radar Hardware Accelerator is included as part of a single chip along with the mmWave RF and analog front end. In this device, two of the accelerator local memories, namely ACCEL_MEM0 and ACCEL_MEM1, are directly shared with the ping and pong ADC buffers (which are 16KB each) – such that the ADC output samples for first-dimension FFT processing are directly and immediately available to the Radar Hardware Accelerator at the end of each chirp, without needing a DMA transfer. After the first-dimension FFT processing is complete (typically, at the end of the active transmission of chirps in a frame), it is possible to freely use these memories for second-dimension FFT processing by bringing in data to these memories through DMA transfer.

The purpose behind the four separate local memories (16KB each) inside the Radar Hardware Accelerator is to enable the *ping-pong* mechanism, for both the input and output, such that the DMA write (and read) operations can happen in parallel to the main computational processing of the accelerator. The presence of four memories enables such parallelism. For example, the DMA can be configured to write FFT input samples (ping) into ACCEL_MEM0 and read FFT output samples (ping) from ACCEL_MEM2. At the same time, the accelerator engine can be working on FFT input samples (pong) from ACCEL_MEM1 and writing FFT output samples (pong) into ACCEL_MEM3. However, both the DMA and the accelerator cannot access the same 16KB

memory at the same time. This would lead to an error (refer to the STATERRCODE register description in Output formatter registers.

The Radar Hardware Accelerator and the main processor (Cortex-M4) in the mmWave xWRLx432 device operate on a single clock domain and the operating clock frequency is 80 MHz.

The accelerator local memories are 128-bits wide, for example, each of the 16KB banks is implemented as 1024 words of 128 bits each. This allows the DMA to bring data into the accelerator local memories efficiently (up to a maximum throughput of 64 bits per clock cycle, depending upon the DMA configuration). Two ports for accessing the HWA local memories are available, and these map the same 64KB into two different address spaces.

It is important to note that any of the four local memories can be the *source* of the input samples to the accelerator engine and any of the four local memories can be the *destination* for the output samples from the accelerator engine – with the important restriction that the source and destination memories cannot be the same 16KB bank. Note also that the accelerator local memories do not necessarily need to be used in ping-pong mode and can instead be used as larger 32KB input and output memories, if the use case requires. The address space for the four 16KB memories is contiguous and thus the source and destination memory can effectively be larger than 16KB.

10.1.1.3.2 Configuration

The operations of the Radar Hardware Accelerator are configured using registers, which are of two types – parameter sets and common (common for all parameter sets) registers. The purpose of the parameter sets is to enable a complete sequence of various accelerator operations to be preprogrammed (with appropriate source and destination memory addresses and other configurations specified for each operation in that sequence), such that the accelerator can perform them one after the other, with minimal intervention from the main processor.

The parameter-set register configurations are programmed into a separate 1024-byte *parameter -setconfiguration memory*. A state machine built into the accelerator handles the loading of one parameter-set configuration at a time and sequences the preprogrammed operations one after another. This process is further explained in later sections of this user's guide.

10.1.1.4 Accelerator Engine Block Diagram

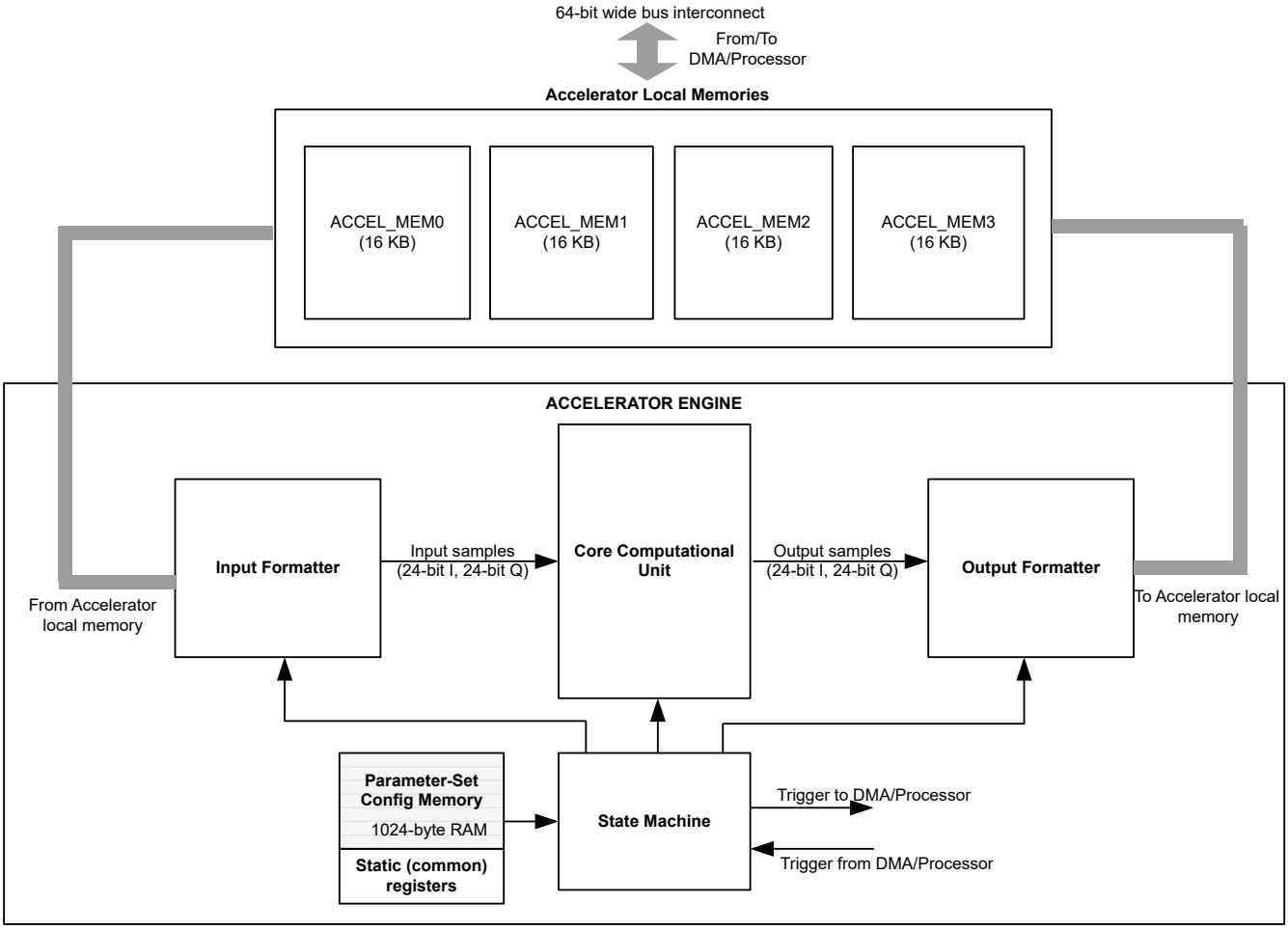


Figure 10-2. AcceleratorEngine Block Diagram

The purpose of these components is as follows:

- State machine: the state machine is responsible for controlling the overall operation of the accelerator – specifically, the starting, looping, stopping, as well as triggering and handshake mechanisms between the accelerator, DMA, and main processor. The state machine is also closely connected to the parameter-set configuration memory and takes care of sequencing and chaining a sequence of multiple accelerator operations as programmed in the parameter-set configuration memory.
- Input formatter: the input formatter block is responsible for reading the input samples from any one of the local memories and feeding them into the core computational unit. In this process, this block provides flexible ways of accessing the input samples, in terms of 16-bit versus 32-bit aligned input samples, transpose read-out, flexible scaling, and sign extension to generate internal bit-width of 24 bits, and so on. Lastly the input formatter block provides 24-bit complex samples as input to the core computational unit. The local memory (memories) from which the input formatter reads the input samples is called the *source* memory.
- Output formatter: the output formatter block is responsible for writing the output samples from the core computational unit into the local memories. This block also provides flexible ways of formatting the output samples, in terms of 16-bit versus 32-bit aligned output samples, transpose write, flexible scaling from internal bit-width of 24 bits, to 16-bit or 32-bit aligned output samples, sign-extension, and so on. The local memory (memories) to which the output formatter writes the output samples is called the *destination* memory.
- Core computational unit: the core computational unit contains the main computational logic for various operations, such as windowing, FFT, magnitude, log2, and CFAR calculations. The unit accepts a streaming input from the input formatter block (at the rate of one input sample per clock cycle), performs computations, and produces a streaming output to the output formatter block (typically at the rate of one output sample per clock cycle), with some initial latency depending on the nature of the computations involved.

- Parameter-set configuration memory: this is a 1024-byte RAM that is used to preconfigure the sets of parameters (register settings) for a chained sequence of accelerator operations, which can then be executed by the state machine in a loop. This allows the accelerator to perform a preprogrammed sequence of operations in a loop without frequent intervention from the main processor.
- The number of parameter sets that can be preconfigured and sequenced (chained) is 32. This means that up to 32 accelerator operations can be chained together and these can then be looped as well, with minimal intervention from the main processor. For example, operations like FFT, log-magnitude, and CFAR-CA/OS detection can be preconfigured in the parameter-set configuration memory and the state machine can be made to sequence them one after another and run them in a loop for specified number of times. There is a provision available to interrupt the main processor and/or trigger a DMA channel at the end of each parameter set if required. This allows various ways by which the accelerator, DMA, and the main processor can work together to establish a data and processing flow. As shown in Figure 10-3, each parameter set contains the equivalent of eight 32-bit registers, which corresponds to total RAM size of 32 × 8 × 32 bits = 1024 bytes for the parameter-set configuration memory.
- The layout of the parameter-set register map is provided in Appendix A. The detailed descriptions of the registers are provided in the various sections, as and when the functionality of each component is presented.

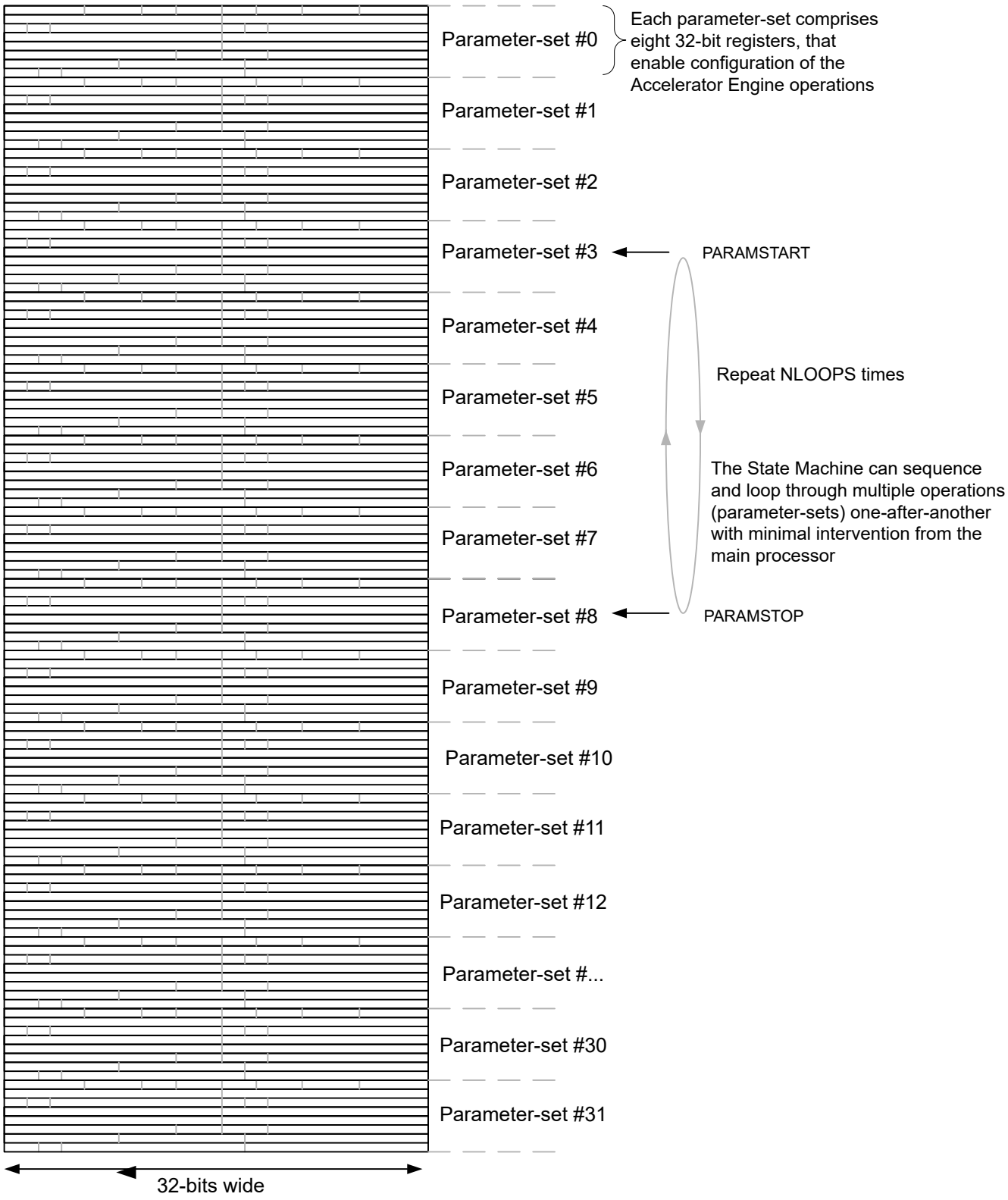


Figure 10-3. Parameter-Set Configuration Memory (1024 Bytes)

10.1.1.5 Accelerator Engine Operation

The accelerator engine and the local memories run on a single clock domain. The overall operation of the accelerator can be summarized as follows. The accelerator engine is configured by the main processor through common configuration registers (common for all parameter sets), as well as the parameter-set configuration

memory. As explained earlier, the former comprises common register settings for overall control of the accelerator engine, and the latter comprises the 32 parameter-set specific settings which control the functioning of the accelerator for each of its chained sequence of operations.

When the accelerator engine is enabled, the state machine kicks off and controls the overall operation of the accelerator, which involves loading the parameter sets one at a time from the parameter-set configuration memory into various internal registers of the accelerator engine and running the accelerator as per the programmed configuration for each parameter set one after another. The entire procedure then repeats in a loop for a programmed number of times (NLOOPS described later).

Each parameter set includes various configuration details such as the accelerator mode of operation (FFT, Log2, and so on), the source memory address, number of samples, the destination memory address, input formatting, output formatting, trigger mode for controlling the start of computations to ensure proper handshake with the DMA, and so on.

10.1.1.5.1 Data Throughput

Once the state machine has loaded the registers corresponding to the current parameter set to be executed, the data flow happens as follows: at each clock cycle, one sample from the source memory is read by the input formatter and fed into the core computational unit with appropriate scaling and formatting as configured. The data interface between the input formatter and the core computational unit is a 24-bit complex bus (24-bit for each I and Q) which streams one input sample every clock cycle. The core computational unit processes this streaming sequence of input samples and in general, produces a streaming output also at one sample every clock cycle, after an initial latency period. Thus for most operations (FFT, log-magnitude, CFAR-CA, and so on), in steady state the core computational unit maintains a streaming data rate of one sample per clock cycle. The data interface between the core computational unit and the output formatter is also a 24-bit complex bus (24-bit for each I and Q) and the output formatter is responsible for writing into the destination memory, with appropriate scaling and formatting as configured.

The next section provides more details regarding the state machine, including its detailed operation, registers, trigger mechanisms, and so on.

10.1.2 Accelerator Engine – State Machine

This section describes the state machine block present in the accelerator engine (see Figure 10-4). This block, together with the input formatter and output formatter blocks described in the next two sections, provides the overall framework for establishing the data flow and using the accelerator for various computations.

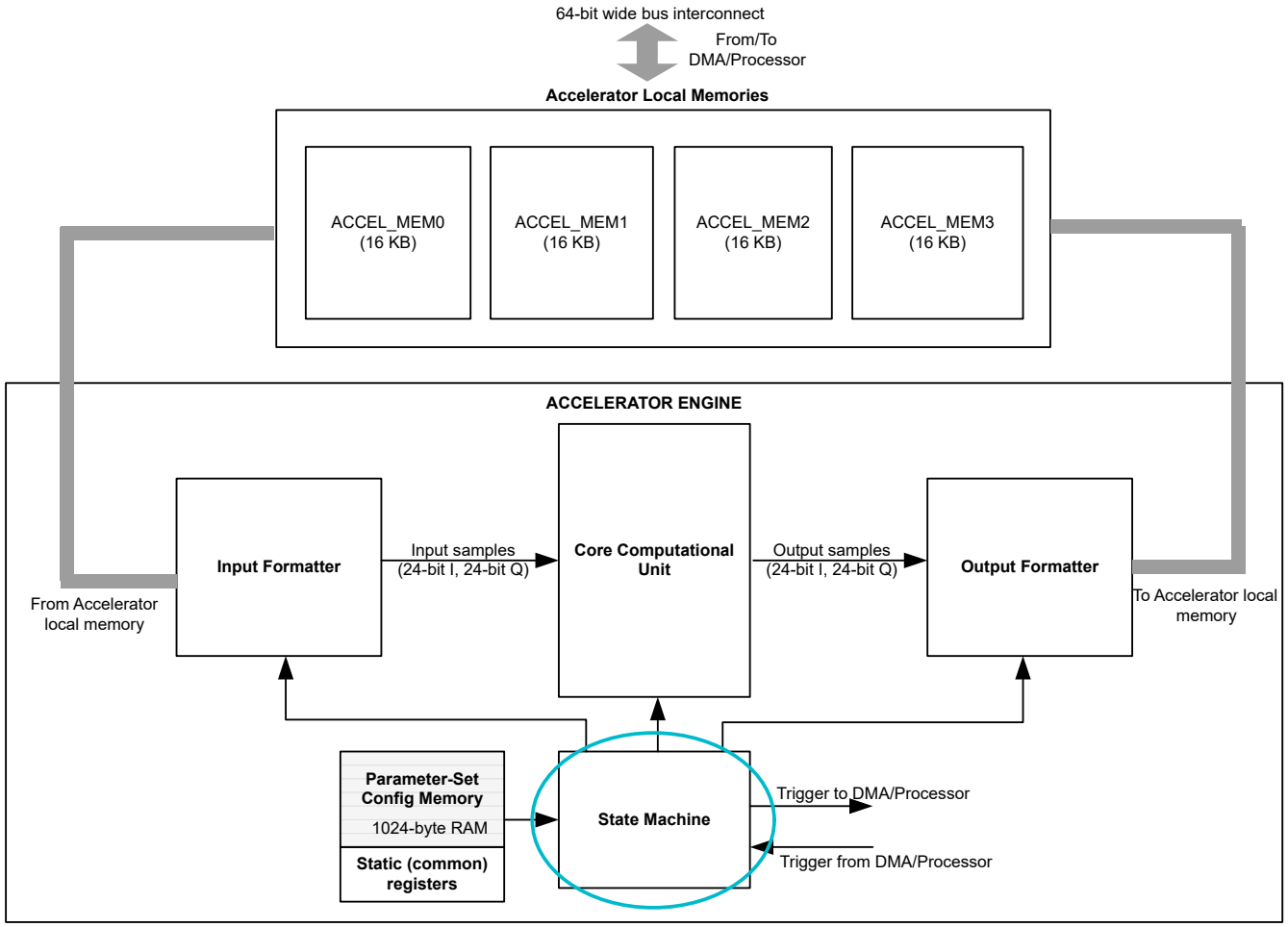


Figure 10-4. State Machine

10.1.2.1 State Machine

The state machine controls the overall functioning of the Radar Hardware Accelerator. The state machine controls the enabling and disabling of the accelerator, as well as supports sequencing an entire set of operations (configured using parameter-set configuration memory), and looping through those operations one after another without needing frequent intervention from the main processor.

10.1.2.1.1 State Machine – Operation

The state machine block and the entire accelerator remain in reset and disabled state by default. The state machine (and hence the accelerator in general) is enabled by setting the ACCCLKEN register bit, followed by writing 111b into the ACCENABLE register.

Note that a complete list of registers pertaining to the state machine is provided in Table 1. Some of the registers are common (common for all parameter sets) registers, whereas some other registers are parameter-set registers, which as explained in the previous section means that they can be uniquely programmed for each of the 32 parameter sets. For each register, Table 1 lists whether it is part of the parameter set or not. Table 1 also provides a brief description of each register.

When enabled, the state machine steps through (one after another) the parameter sets programmed in the parameter-set configuration memory and executes the computations as per the configuration of each parameter set. The registers PARAMSTART and PARAMSTOP define the starting index and ending index within the 16 parameter sets, so that only those parameter sets between the start and end indices are executed by the accelerator, as shown in Figure 10-3. The state machine also loops through these parameter sets for a total of NLOOPS times (unless NLOOPS is programmed as 0 or 4095, in which case the loop does not run or runs infinite times respectively). As an example, if the state machine needs to be configured to run the first four

parameter sets in a loop 64 times, then the registers should be programmed as follows: PARAMSTART = 0, PARAMSTOP = 3, and NLOOPS = 64.

For each parameter set, there is a TRIGMODE register, which is used to control when the state machine starts executing the computations for that parameter set. This control is useful, for example, to ensure that the input data is ready in the accelerator local memory (source memory) before the computations are started. Specifically, it is possible to trigger the start of computations after completion of a DMA transfer, or, after a ping-pong switch happens in the ADC buffer, and so on. The TRIGMODE register setting thus controls when the accelerator operation is triggered for the current parameter set and there are four trigger mechanisms supported as listed in the next subsection. Once triggered, the state machine loads all the registers from the parameter-set configuration memory for the current parameter set into corresponding internal registers of the accelerator and starts the actual computations for that parameter set. After completion of computations of the current parameter set, it moves to the next parameter set.

After a sequence of operations as programmed in the parameter set(s) for the specified number of loops is complete, the accelerator provides a completion interrupt (ACC_DONE_INTR) to the processor. The accelerator can be reconfigured as desired. For reconfiguration, the following procedure must be followed. The accelerator must be disabled by writing 000b to the ACCENABLE register. Then, a reset must be asserted by writing 111b followed by 000b to the ACCRESET register. The new configurations can now be written in to the accelerator, and then the accelerator can be enabled again by writing 111b to ACCENABLE.

10.1.2.1.2 State Machine – Trigger Mechanisms (Incoming)

As mentioned in the previous subsection, for each parameter set, the start of the computations can be triggered based on specific events. Four trigger mechanisms are supported as follows:

- Immediate trigger (TRIGMODE = 000b): In this case, the state machine does not wait for any trigger and starts the accelerator computations immediately for the current parameter set. This mode is applicable when chaining (sequencing) a set of operations one after another in the accelerator without any need for control handshake or data exchange outside the accelerator (for example, when chaining FFT and log-magnitude operations) with no need to wait for a trigger in between.
- Wait for processor-based software trigger (TRIGMODE = 001b): This is a software-triggered mode that is useful when the main processor must directly control the data flow and start or stop of accelerator computations. In this trigger mode, the state machine waits for a software-based trigger, which involves the main processor setting a separate self-clearing bit in a CM42ACCTRIG register (single-bit register). The state machine keeps monitoring that register bit and waits as long as the value is zero. When the value becomes 1 (set), the state machine gets triggered to start the accelerator operations for the current parameter set.
- Wait for the ADC buffer ping-to-pong or pong-to-ping switch (TRIGMODE = 010b): This trigger mode is specific to the mmWave xWRL6432/xWRL1432 device, which has RF and analog front end integrated in the same chip with the main processor and the Radar Hardware Accelerator. Recall that in the mmWave xWRL6432/xWRL1432 device, the ADC ping and pong buffers are shared with the accelerator local memories (ACCEL_MEM0 and ACCEL_MEM1), such that the ADC data is directly available to the accelerator for processing during active chirping portion of the frame. This sharing mode is enabled by setting the FFT1DEN register bit before the start of the frame. In this trigger mode, the state machine of the accelerator starts the computations for the current parameter set as soon as the ADC buffer switches from ping-to-pong or pong-to-ping. As an example, during the active chirping portion of a frame, the mmWave xWRL6432/xWRL1432 digital front end and ADC buffer can be configured to switch from ping-to-pong or pong-to-ping buffer at the end of every chirp or at the end of every few chirps or at the end of every specified number of ADC samples. These xWRL6432/xWRL1432 digital front-end configurations are accomplished using other registers unrelated to the Radar Hardware Accelerator and not described in this document. Now, using this trigger mode (TRIGMODE = 010b) allows the accelerator computations to start whenever the ping-to-pong or pong-to-ping switch happens in the ADC buffer, thus enabling inline per- chirp processing. It is important to mention here that the user must take care to ensure that processing of the current ping data is completed by the accelerator, before the next switch/trigger happens on the ADC buffer. In other words, the chirp duration (ping-pong switch frequency) must be configured to be so fast that the accelerator cannot complete its configured operations within that duration
- Wait for the DMA-based trigger (TRIGMODE = 011b): This trigger mode is useful when a DMA transfer completion must be used to trigger the start of the accelerator computations for the current parameter set. The primary purpose of this trigger mode is as follows; when performing second dimension FFT, the DMA is

used to bring the FFT input samples from the Radar data memory to the local memory of the accelerator. Upon completion of each DMA transfer, it is useful to automatically trigger the accelerator to perform the FFT. To achieve this, the state machine of the accelerator has a 16-bit register called the DMA2ACCTRIG register, where each register bit maps to one of 16 DMA channels that are associated with the accelerator. To use the DMA-based trigger mode, the DMA2ACC_CHANNEL_TRIGSRC register in the current parameter set must be programmed to the DMA channel whose completion we wish to monitor. The state machine then monitors the corresponding register bit in the DMA2ACCTRIG register, and triggers the execution of the current parameter set only when that register bit gets set. For e.g. if DMA2ACC_CHANNEL_TRIGSRC is programmed to 5, then the current parameter set will execute only once the register bit #5 gets set in DMA2ACCTRIG. The user may utilize the EDMA's linking capability to set the appropriate register bit in DMA2ACCTRIG. Linking is a programmable feature of the EDMA, where the completion of a DMA transfer can automatically trigger a second DMA transfer. In the present context, the DMA transfer that moves data to the local memory of the accelerator can be linked to a second DMA whose purpose is to write a one-hot signature into DMA2ACCTRIG to set a specific register bit and trigger the accelerator. Note that there are 16 read-only, one-hot, signature registers (SIG_DMACH1_DONE, SIG_DMACH2_DONE, and more) that are available. These registers are simply read-only registers which contain hard-coded values (each register is a one-hot signature – 0x0001, 0x0002, 0x0004, 0x0008, and so on). For convenience, these hard-coded 16 read-only signatures can be used, so that the second DMA can simply copy from one of these SIG_DMACHx_DONE registers into the DMA2ACCTRIG register to set the appropriate register bit.

10.1.2.1.3 State Machine – Trigger Mechanisms (Outgoing)

After the accelerator computations for the current parameter set are triggered (using one of the four incoming trigger mechanisms mentioned in the previous subsection), it performs the actual computation operations for that parameter set. These computations typically take several tens or hundreds of clock cycles, depending on the nature of the configuration programmed. Once the accelerator completes its computation operations for the current parameter set, the state machine advances to the next parameter set and repeats the same process. But before advancing to the next parameter set, it can interrupt the main processor and/or trigger a DMA channel. This provision is useful if the main processor is required to read or write registers or memory locations at the end of the current parameter set. Also, this provision is useful for triggering a DMA channel, so that the output of the accelerator can be copied out of the accelerator local memories.

There are two trigger mechanisms provided as follows:

- Interrupt to main processor (CM4INTREN = 1): The accelerator interrupts the main processor at the end of completion of computations for the current parameter set, if the register bit CM4INTREN is set.
- Trigger to DMA (DMATRIGEN = 1): The accelerator gives a trigger to a DMA channel at the end of completion of computations for the current parameter set, if the register bit DMATRIGEN is set. If DMATRIGEN is set, then the particular DMA channel as specified in a separate ACC2DMA_CHANNEL_TRIGDST register (valid values are 0 to 15, for the 16 DMA channels dedicated for the accelerator) is triggered. Thus, it is possible to preconfigure up to 16 DMA channels and trigger the appropriate one at the end of the computations of the current parameter set. The trigger from accelerator to the DMA channels can also be faked by the processor, by writing to a CM42DMATRIG register.

This can be used by the processor to kick-start a full/repetitive chain of operations, that are then subsequently managed between the DMA and the accelerator without further processor involvement – for example, the processor writes to the CM42DMATRIG register to trigger a DMA channel for the first time, and this kicks off a series of back-to-back data transfers and accelerator computations, with the DMA and accelerator hand-shaking with each other.

10.1.2.1.4 State Machine – Register Descriptions

Table below lists all the registers of the state machine block. As explained previously, some of the registers are common (common for all parameter sets) registers, whereas some others are *partof each parameter set*. For each register, this distinction is captured as part of the register description in Table 1.

Table 10-1. State Machine Registers

Register	Width	ParameterSet	Description
ACCENABLE (see Table 10-17.)	3	No	Enable andDisable Control: This registerenables or disables the entire Radar Hardware Accelerator. The reason for a 3-bit register (instead of 1-bit) is to avoid an accidental bit-flip (for example, transient error caused by a neutron strike) from unintentionally turning on the accelerator engine. A value of ACCENABLE = 111b enables the Radar Hardware Accelerator and any other value of the register keeps the accelerator engine in disabled state.
ACCCLKEN (see Table 10-17.)	1	No	Clock-gatingControl: This registerbit controls the enable/disable for the clock of the Radar Accelerator. This register bit can be set to 0 to clock-gate the accelerator when not using the accelerator. Before enabling the accelerator or before configuring the accelerator's registers, this register bit should be set first, so that the clock is available.
ACCRESET (see Table 10-17.)	3	No	Software ResetControl: This registerprovides software reset control for the Radar Hardware Accelerator. The assertion of these register bits by the main processor will bring the accelerator engine to a known reset state. This is mostly applicable for resetting the accelerator in case of unexpected behavior. Under normal circumstances, it is expected that whenever the accelerator is enabled (from disabled state), it always comes up in a known reset state automatically. The recommended sequence to be followed in case software reset is desired is to write 111b to this register and then a 000b, before the clock is enabled to the accelerator.
NLOOPS (see Table 10-92.)	12	No	Numberof loops: This registercontrols the number of times the state machine will loop through the parameter sets (from a programmed start index till a programmed end index) and run them. The maximum number of times the loop can be made is run is 4094. A value of 4095 (0xFFF) programmed in this register should be considered as a special case and it should be interpreted as an infinite loop mode, for example, keep looping and never stop the accelerator engine unless reset by the main processor. A value of zero programmed in this register means that the looping mechanism is disabled. In this case, the accelerator engine can still be used under direct control of the main processor (without the state machine looping provision coming into the picture).
PARAMSTART (see Table 10-92.)	5	No	Parameter-set Start andStop Index: These registersare used to control the start and stop index of the parameter set through which the state machine loops through. The state machine starts at the parameter set specified by PARAMSTART and loads each parameter set one after another and runs the accelerator as per that configuration. When the state machine reaches the parameter set specified by PARAMSTOP, it loops back to the start index as specified by PARAMSTART.
PARAMSTOP (see Table 10-92.)	5	No	

ADVANCE INFORMATION

Table 10-1. State Machine Registers (continued)

Register	Width	ParameterSet	Description
FFT1DEN (see Table 10-17.)	1	No	ADC buffersharing mode This registeris relevant when the Radar Hardware Accelerator is included in a single device along with the mmWave RF front-end. In such a case, during active chirp transmission and inline first dimension FFT processing, the ACCEL_MEM0 and ACCEL_MEM1 memories of the accelerator are shared as ping-pong ADC buffers. This register bit needs to be set during this time, so that while the digital front end writes ADC samples to the ping buffer, the accelerator automatically accesses (only) the pong buffer, and vice versa. At the end of the active transmission portion of a frame, this bit can be cleared, so that the accelerator has access to all the four local memories independently.
TRIGMODE	3	Yes	Trigger modecontrol: This parameter-setregister is used to control how the state machine and the operations of the accelerator are triggered for each parameter set. The following modes are supported: • 000b– Immediate trigger • 001b– Software trigger • 010b– Ping-pong switch based trigger (applicable only when FFT1DEN is set) • 011b– DMA-based trigger The trigger modesare described in Section 2.1.2.
CM42ACCTRIG (see Table 10-19.)	1	No	Software trigger bit: This registerbit is relevant whenever software triggered mode is used (for example, TRIGMODE = 001b). Whenever software triggered mode is configured for a parameter set, the state machine keeps monitoring this register bit and waits as long as the value is zero. The main processor software can set this register bit, so that the state machine gets triggered and starts the accelerator operations for that parameter set.
DMA2ACCTRIG (see Table 10-77.)	16	No	DMAtrigger register: This registeris relevant whenever DMA triggered mode is used (for example, TRIGMODE = 011b). Whenever a DMA channel has finished copying input samples into the local memory of the accelerator and wants to trigger the accelerator, the procedure to follow is to use a second linked DMA channel to write a 16-bit one-hot signature into this register to trigger the accelerator. In DMA triggered mode, the state machine keeps monitoring this 16-bit register and waits as long as a specific bit (see DMA2ACC_CHANNEL_TRIGSRC) in this register is zero. The second linked DMA channel writes a one-hot signature that sets the specific bit, so that the state machine gets triggered and starts the accelerator operations for that parameter set.
DMA2ACC_CHANNEL_ TRIGSRC	4	Yes	DMAchannel select for DMA completion trigger: This parameter-setregister is relevant whenever DMA triggered mode is used (for example, TRIGMODE = 011b). This register selects the bit number in DMA2ACCTRIG for the state machine to monitor to trigger the operation for that parameter set.

Table 10-1. State Machine Registers (continued)

Register	Width	ParameterSet	Description
CM4INTREN	1	Yes	Completioninterrupt to main processor: This parameter-setregister is used to enable/disable interrupt to the main processor upon completion of the accelerator operation for that parameter set. If enabled, the main processor receives an interrupt from the Radar Hardware Accelerator at the end of operations for that parameter set, so that the main processor can take any necessary action.
PARAMDONESTAT (read-only) (see Table 10-28.)	32	No	Parameter-set donestatus: This read-onlystatus register can be used by the main processor to see which parameter sets are complete that led to the interrupt to the main processor. The individual bits in this 16-bit status register indicate which of the 16 parameter sets have completed. These status bits are not automatically cleared, but they can be individually cleared by writing to another 16-bit register PARAMDONECLR.
PARAMDONECLR (see Table 10-29.)	32	No	
DMATRIGEN	1	Yes	Completiontrigger to DMA: This parameter-setregister is used to enable DMA channel trigger upon completion of the accelerator operation for that parameter set. This trigger mechanism enables the accelerator to hand-shake with the DMA so that output data samples are copied out of the accelerator local memory. If enabled, the accelerator triggers a specified DMA channel, so that the output samples can be shipped from the local memory to Radar data memory.
ACC2DMA_CHANNEL_ TRIGDST	4	Yes	DMAchannel select for accelerator completion trigger: This parameter-setregister is used to select which of the 16 DMA channels allocated to the accelerator should be triggered upon completion of the accelerator operation for that parameter set. This register is to be used in conjunction with DMATRIGEN.
CM42DMATRIG (see Table 10-19.)	16	No	Trigger from processorto DMA: This registercan be used by the processor to trigger a DMA channel for the first time, so that a full sequence of repeated operations between the DMA and the accelerator gets kick-started.
PARAMADDR (see Table 10-56.)	5	No	Debugregister for current parameter-set index: This read-onlystatus register indicates the index of the current parameter set that is under execution. This is useful for debug, where parameter sets can be executed in single-step manner (one-by-one) using SW trigger mode for each of them. In such a debug, this register indicates which parameter set is currently waiting for the SW trigger.
LOOPCNT (see Table 10-56.)	12	No	Debugregister for current loop count: This read-onlystatus register indicates what is the loop count that is presently running. When the state machine is programmed for NLOOPS loops, this register shows the current loop count that is running.

Table 10-1. State Machine Registers (continued)

Register	Width	ParameterSet	Description
ACC_TRIGGER_IN_STAT (see Table 10-26.)	19	No	Debugregister for trigger status: This isa read-only status register, which indicates the trigger status of the accelerator, for example, whether a specific DMA trigger or a Ping-pong trigger or a SW trigger was ever received (refer TRIGMODE). The MSB 16 bits of this register indicate whether a trigger was received via DMA trigger method. The next two bits (for example, bit indices 2 and 1) indicate the status of DFE ping-pong switch-based trigger and SW trigger respectively. The LSB bit is always 1 and can be ignored.
ACC_TRIGGER_IN_CLR (see Table 10-26.)	1	No	Cleartrigger status read-only register: This register-bitwhen set clears the trigger status register ACC_TRIGGER_IN_STAT described above.

The next two sections cover the Input Formatter and Output Formatter blocks, including their detailed operation, registers and usage procedure.

10.1.3 Accelerator Engine – Input Formatter

This section describes the input formatter block present in the accelerator engine (see Figure 10-5).

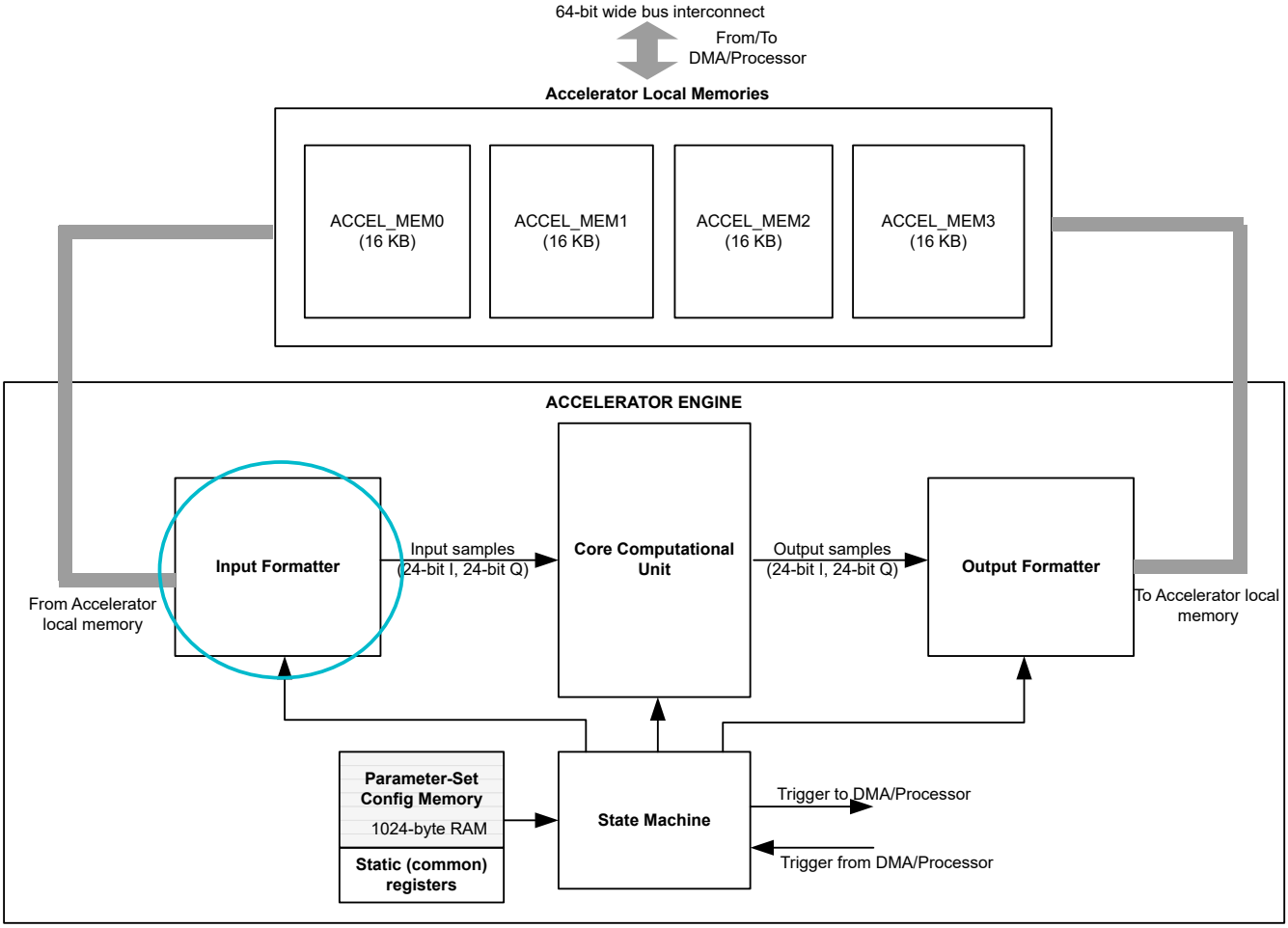


Figure 10-5. Input Formatter

ADVANCE INFORMATION

10.1.3.1 Input Formatter

The input formatter is used to access, format, and feed the data from the local memories of the accelerator as 24-bit I and 24-bit Q samples into the core computational unit. The input formatter provides various capabilities to access and format the samples from the local memories – especially, various multidimensional access patterns (for example transpose access), 16-bit or 32-bit aligned word access, scaling using bit-shifts to generate 24-bit wide samples from 16-bit or 32-bit words, real versus complex input, sign extension, conjugation, and more.

10.1.3.1.1 Input Formatter – Operation

The input formatter block is responsible for reading the input samples from the accelerator local memory and feeding them into the core computational unit (see Figure 10-2). The data flow from the input formatter, through the core computational unit, to the output formatter is designed to sustain a steady-state throughput of one complex sample per clock cycle. The input formatter thus feeds one sample (24-bit I and 24-bit Q) into the core computational unit every clock cycle.

To make the best use of the capabilities of the core computational unit and to allow meaningful chaining of radar signal processing operations with minimal intervention from the M4 processor, the input formatter supports flexibility in how the input samples are accessed from the memory and how they are formatted and fed into the core computational unit.

The memory from which the input formatter picks up the data is referred to as *sourcememory*. Note that any of the four accelerator local memories can be the source memory. However, as will be described in a subsequent section, there is an important restriction which explains that the source memory cannot be the same as the destination memory (which is the memory to which the output formatter writes the output data).

10.1.3.1.2 Input Formatter – 2D Indexed Addressing for Source Memory Access

The 16-bit parameter-set register SRCADDR specifies the start address at which the input samples must be accessed. This register is a byte-address, and a value of 0x0000 corresponds to the first memory location of ACCEL_MEM0 memory. The 16-bit SRCADDR register maps to the entire 64KB address space of the four accelerator local memories (4x16KB).

The input data can be read from the memory as either 16-bit wide samples or 32-bit wide samples. Also, they can be read as real samples or complex samples. These two aspects are configured using register bits SRC16b32b and SRCREAL. See Table 2 for a description of these and other registers pertaining to the input formatter block. As an example, if SRC16b32b = 0 and SRCREAL = 0, then the input samples are read from the memory as 16-bit complex samples (16-bit I and 16-bit Q), shown in Figure 10-6.

An important feature of the input formatter block is that it supports flexible access pattern to fetch data from the source memory, which makes it convenient when the data corresponding to multiple RX channels are interleaved or when performing multi-dimensional (FFT) processing. This feature is facilitated through the SRCAINDX, SRCACNT, SRCBINDX, and REG_BCNT registers, which are part of each parameter-set configuration.

The register SRCAINDX specifies how many bytes separate successive samples to be fetched from the source memory and the register SRCACNT specifies how many samples need to be fetched per iteration. An iteration is typically one computational routine, such as one FFT operation. It is possible to perform multiple iterations back-to-back – for example, four FFT operations corresponding to four RX channels.

The register SRCBINDX specifies how many bytes separate the start of input samples for successive iterations and REG_BCNT specifies how many iterations to perform back-to-back. These registers can be better understood using the example given in Figure 10-6. Also, a complete use case is illustrated in

Section 6, which provides further clarity on this aspect. In Section 6, the input data consists of complex data (16-bit I and 16-bit Q) that is contiguously present in ACCEL_MEM0. The data in memory consists of four sets of 128 samples each (say, corresponding to four RX antennas) and these are shown in four different colors. Because each sample occupies 4 bytes and the samples are contiguously placed in the memory starting at the beginning of ACCEL_MEM0, values of SRCADDR = 0x0000 and SRCAINDX = 4 are used to fetch these samples.

In each clock cycle, the input formatter fetches one complex sample from the memory and feeds it into the core computational unit (with appropriate scaling, as described later). Because there are 128 samples to be fed for the first iteration (computational routine), a value of SRCACNT = 127 is used. For the second iteration, the samples are fetched starting from a memory location that is SRCBINDX (=128 × 4 = 512) bytes away from SRCADDR.

This process repeats for the programmed number of iterations as per the REG_BCNT register. For example, the value of REG_BCNT = 3 used in this example corresponds to four iterations. Note that the registers shown here are part of parameter-set configuration registers and the four iterations described here can be performed using a single parameter set.

Figure 10-6. Input Formatter Source Memory Access Pattern (Example)

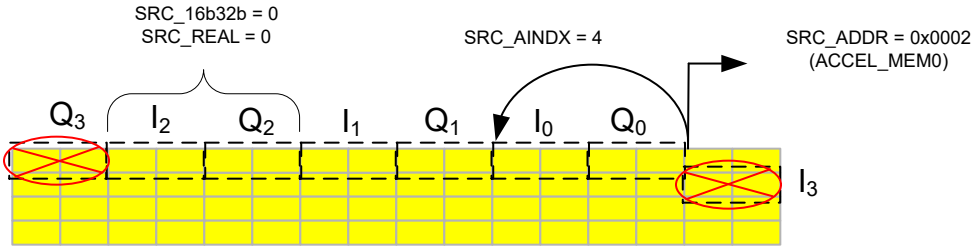
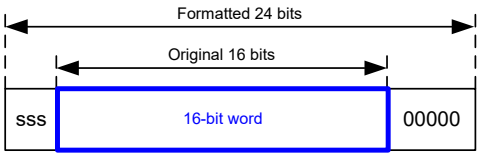


Figure 10-7. Invalid Configuration Example

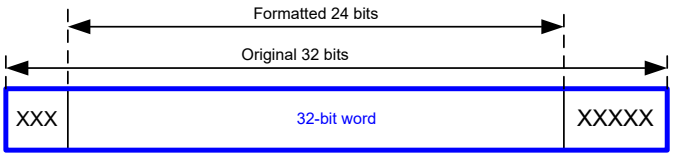
An important restriction in programming the registers related to source memory access pattern is that the input formatter can only read data from one memory row (128-bit memory location) in a clock cycle.

Therefore, if a sample is placed in memory such that the real-part (I value) is at the end of one memory location and the imaginary part (Q value) is at the beginning of the next memory location, then that would be an invalid configuration (see Figure 10-7).

10.1.3.1.3 Input Formatter – Scaling and Formatting



For 16-bit case, if SRC_SCAL = 3, then 5 zeros are padded at the LSB, and 3 redundant (extension) bits are padded at the MSB



For 32-bit case, if SRC_SCAL = 5, then 5 bits are dropped at the LSB, and 3 bits are clipped (with saturation) at the MSB

Figure 10-8. Input Formatter Data Scaling

The input formatter allows the input samples read from the source memory to be scaled and formatted before feeding them as 24-bit complex samples into the core computational unit.

Even though the data read from the source memory is initially 16-bits or 32-bits wide (for each I and Q), the samples expected by the core computational unit are 24-bit complex samples (24-bits each for I and Q). There is a REG_SRCSCAL register which provides scaling options using bit-shift to generate 24-bit samples from the original 16- or 32-bit data (see Figure 10-8)

For the 16-bit case, the 24-bit sample is generated by padding (8-REG_SRCSCAL) zeros at the LSB and REG_SRCSCAL redundant MSBs. For the 32-bit case, the 24-bit sample is generated by dropping REG_SRCSCAL bits at the LSB and clipping (8-REG_SRCSCAL) bits at the MSB. Note that the register bit SRCSIGNED is used to indicate whether the input samples are signed or unsigned. When this register bit is set, the input samples are treated as signed numbers and hence any extra MSB bits are sign- extended and any clipping of MSB bits takes care of signed saturation. In most cases of interest in part one of this user guide (for example, when performing FFT operation), the input samples would be signed and hence SRCSIGNED should be set (for example, equal to 1).

When the input samples are complex (for example, SRCREAL = 0), there is a provision to conjugate the input samples. Setting the register bit SRCCONJ conjugates the input samples before feeding them to the core computational unit. This feature (together with a corresponding DSTCONJ register bit in the output formatter block) enables an IFFT mode from the FFT engine. Note that conjugating the input and output of an FFT block is equivalent to an IFFT function.

There are other registers in the input formatter, such as BPM_EN, BPMPATTERNLSB and BPMPATTERNMSB, BPMRATE, CIRCIRSHIFT, CIRCSHIFTWRAP, and so on, which are beyond the scope of part one of this user's guide and these registers are described in part two. For the immediate purpose of the first part of the user's guide, it is important to note that BPM_EN and CIRCIRSHIFT registers must be kept 0.

10.1.3.1.4 Input Formatter – Zero Padding

The input formatter has provision for zero padding, which is important when performing FFT of a set of samples whose length is not a power of 2. The input formatter automatically feeds the required number of zeros into the core computational unit, whenever the FFT size (as programmed using the FFTSIZE register, which is described in a later section) does not match the SRCACNT setting.

For example, if the number of input samples read by the input formatter is 56 (for example, SRCACNT =55) and the FFT size is programmed to be 64 (for example, FFTSIZE = 6), then the input formatter feeds 8 zeros at the end of each iteration, before starting to read the input samples for the next iteration from the source memory. This zero-padding provision enables the core computational unit to perform 64-point FFT with the correct set of zero-padded input samples. It is important for the user to note that SRCACNT should never be larger than 2^FFTSIZE-1.

The zero padding is effective only when performing FFT operation in the core computational unit (i.e., when FFT_EN = 1) and not otherwise. Please refer to section 6 for further information regarding the registers relevant for FFT operation.

10.1.3.1.5 Input Formatter – Register Descriptions

Table below lists all the registers of the input formatter block.

Table 10-2. Input Formatter Registers

Register	Width	Parameter Set	Description
SRCADDR	16	Yes	Source startaddress: This registerspecifies the starting address of the input samples, for example, it specifies the source memory start address from which input samples have to be fetched by the input formatter. This is a byte-address and this 16-bit register covers the entire address space of the four local memories (4 × 16KB = 64 KB). The four accelerator local memories are contiguous in the memory address space and any of them can act as the source memory (as long as the same memory bank is not configured to be used as destination memory at the same time).
SRCACNT	12	Yes	Source samplecount: This registerspecifies the number of samples (minus 1) from the source memory to process for every iteration. The sample count is in number of samples, not number of bytes. For example, the sample count can be specified as 255 (SRCACNT = 0x0FF) in a case where a 256-point FFT is required to be performed. Note however that the sample count register does not always match the FFT size. This can happen when zero-padding of input samples is required. For example, a sample count of 192 could be used with an FFT size of 256, in which case, the input formatter will automatically append 64 zeros.
SRCAINDX	16	Yes	Source sampleindex increment: This registerspecifies the number of bytes separating successive samples in the source memory. For example, a value of SRCAINDX = 16 means that successive samples are separated by 16 bytes in memory. The maximum value allowed for this register is 32767.

Table 10-2. Input Formatter Registers (continued)

Register	Width	Parameter Set	Description
REG_BCNT	12	Yes	Numberof iterations: This registerspecifies the number of times (minus 1) the processing should be repeated. This register can be used to process the four RX chains back-to-back – for example, a value of REG_BCNT = 3 means that the processing (say first dimension FFT processing) is repeated four times. Note the distinction between the NLOOPS register of the state machine block and the REG_BCNT register of the input formatter block. The NLOOPS register specifies how many times the state machine loops through all the configured parameter sets (with each time possibly awaiting a trigger), whereas the register REG_BCNT specifies how many times the input formatter and the computational processing of the accelerator is iterated back-to-back for the current parameter set (without any intermediate triggers).
SRCBINDX	16	Yes	Source offsetper iteration: This registerspecifies the number of bytes separating the starting address of input samples for successive iterations. For example, when using four iterations to process the four RX chains, this register can be used to specify the offset in the starting address between the successive RX chains. Note the distinction that SRCAINDX specifies the number of bytes separating successive samples for a particular iteration, whereas SRCBINDX specifies the number of bytes separating the starting address of the first sample for successive iterations. The maximum value allowed for this register is 32767.
SRCREAL	1	Yes	Complexor real Input: This register-bitspecifies whether the input samples are real or complex. A value of SRCREAL = 0 implies complex input and a value of SRCREAL = 1 implies real input. When real input is selected, the input formatter block automatically feeds zero for the imaginary part into the core computational unit.
SRC16b32b	1	Yes	16-bitor 32-bit input word alignment: This register-bitspecifies whether the input samples fetched from source memory are to be read as 16-bits or 32-bits wide. A value of SRC16b32b = 0 implies that the input samples are 16-bits wide each (in case of complex input, real and imaginary parts are each 16 bits wide). A value of SRC16b32b = 1 implies that the input samples are 32-bits wide each.
SRCSIGNED	1	Yes	Input sign-extensionmode: This register-bit,when set, specifies that the input samples are signed numbers and hence, sign-extension or signed-saturation at the MSB is required when converting 16- bit or 32-bit input words to the 24-bit wide samples to be fed into the core computational unit.
SRCCONJ	1	Yes	Input conjugation: This register-bitspecifies whether the input samples should be conjugated before feeding them into the core computational unit. If SRCCONJ is set, then the input samples are conjugated. Setting this register-bit only makes sense if the samples are complex numbers (for example, SRCREAL = 0). This register, together with its counterpart in the output formatter block, enable an IFFT mode for the FFT engine. Note that conjugating the input and output of an FFT block is equivalent to an IFFT function.
REG_SRCSCAL	4	Yes	Input scaling: This registerspecifies a programmable scaling using bit-shift, when converting the 16-bit or 32-bit wide input data to 24-bit wide samples before feeding into the core computational unit. See Figure 10-8 and its description for more details regarding this register.

Table 10-2. Input Formatter Registers (continued)

Register	Width	Parameter Set	Description
CIRCIRSHIFT	–	–	Describedin part two of this user's guide. For the immediate purposes relevant to part one of this user's guide, all of these registers must be kept as 0.

10.1.4 Accelerator Engine – Output Formatter

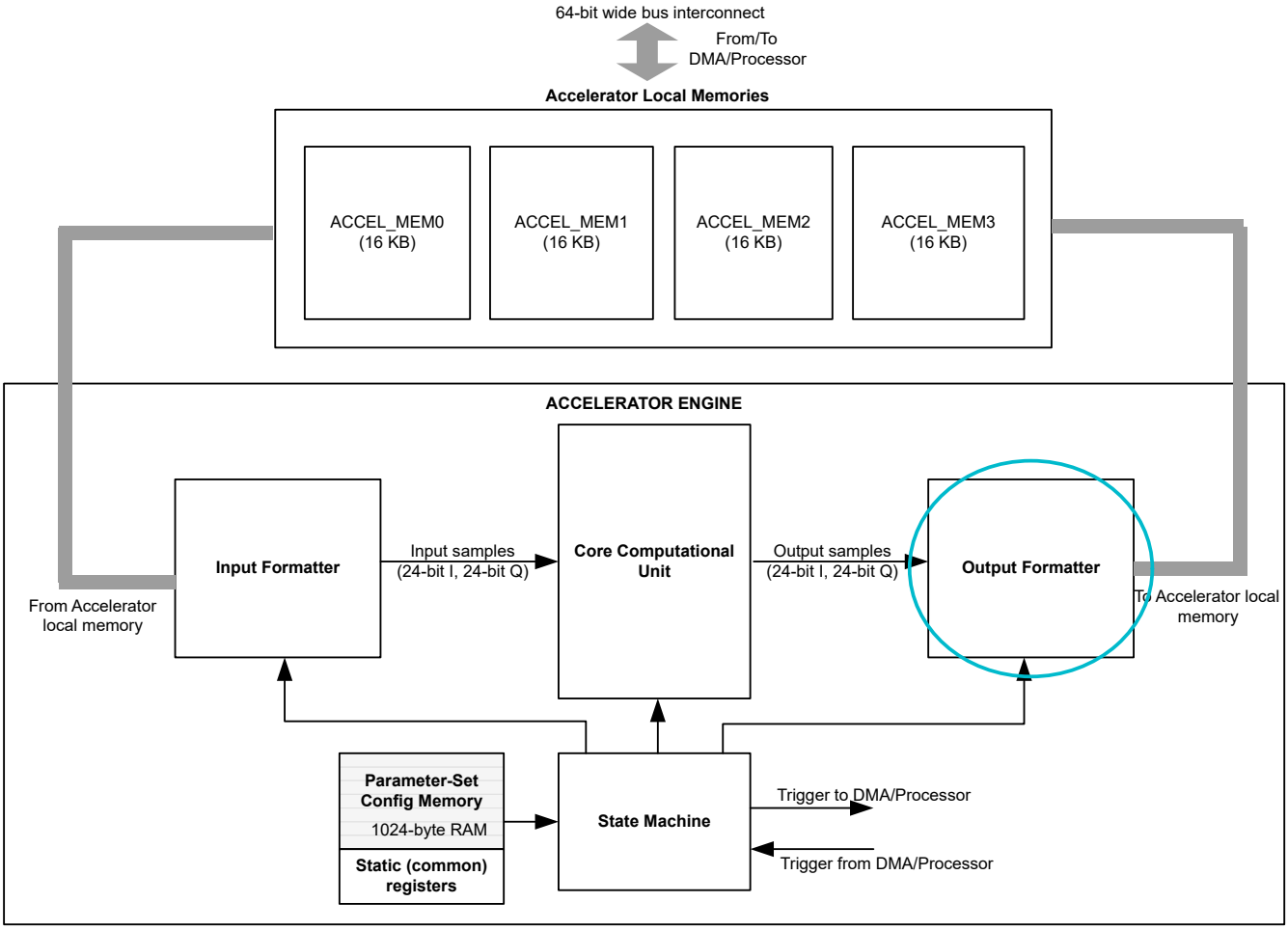


Figure 10-9. Output Formatter

10.1.4.1 Output Formatter

The output formatter is used to format and write the data coming out of the core computational unit into the accelerator local memory. Similar to the input formatter block discussed in the previous section, the output formatter block also provides various capabilities to format and write the samples written to the local memory – especially, various multidimensional access patterns (for example, transpose writes), 16- bit or 32-bit aligned word writes, scaling using bit-shifts to generate 16-bit or 32-bit words from 24-bit wide samples, real versus complex output write, and more.

10.1.4.1.1 Output Formatter – Operation

The output formatter block is responsible for storing the samples coming out of the core computation unit into the accelerator local memory (see Figure 10-2). As mentioned in the previous section, the data flow from the input formatter, through the core computational unit, to the output formatter, is designed to sustain a steady-state throughput of one complex sample per clock cycle. Thus, typically, the output formatter accepts one sample (24-bit I and 24-bit Q) from the core computational unit every clock cycle and writes it to the accelerator local

memory. Just like the input formatter, the output formatter also supports lot of flexibility in how the samples are formatted and written into the memory.

The memory into which the output formatter writes the data is referred to as destination memory. Note that any of the four accelerator local memories can be the destination memory, with the important restriction that the source memory cannot be same as the destination memory. In other words, each of the four 16KB memory banks can either function as source memory, or as destination memory at any time (for example, in any given parameter set).

10.1.4.1.2 Output Formatter – 2-D Indexed Addressing for Destination Memory Access

The 16-bit parameter-set register DSTADDR specifies the start address at which the output samples must be written into the accelerator local memory. Similar to the SRCADDR register of the input formatter, the DSTADDR register of the output formatter is a byte-address and a value of 0x0000 corresponds to the first memory location of ACCEL_MEM0 memory. The 16-bit DSTADDR register maps to the entire 64KB address space of the four accelerator local memories (4 × 16KB). As mentioned in the previous paragraph, in a given parameter set, SRCADDR and DSTADDR cannot be configured such that the input samples being fetched and the output samples being written out are accessing the same memory bank.

Even though the core computational unit produces a 24-bit complex output stream, this output data can be written to the memory as either 16-bit wide samples or 32-bit wide samples. Also, they can be written out as complex samples or real samples (for example, drop imaginary part – applicable when performing log-magnitude computation). These two aspects are configured using register bits DST16b32b and DSTREAL. See Table 3 for a description of these and other registers pertaining to the output formatter block. As an example, if DST16b32b = 0 and DSTREAL = 0, then the output samples are written to the memory as 16- bit complex samples (16-bit I and 16-bit Q), shown in Figure 10-10.

Similar to the input formatter block, the output formatter block also supports flexible patterns to write multidimensional data to the destination memory and this makes it convenient when the data corresponding to multiple RX channels must be interleaved, or when performing multidimensional (FFT) processing. This feature is facilitated through the DSTAINDX, DSTACNT, DSTBINDX, and REG_BCNT registers, which are part of each parameter-set configuration.

The register DSTAINDX specifies how many bytes separate successive samples to be written to the destination memory and the register DSTACNT specifies how many samples must be written per iteration. Note that DSTACNT can be different from SRCACNT – this is useful when only a subset of the output samples need to be stored in the output memory (for example, if some FFT output bins must be discarded). The register DSTBINDX specifies how many bytes separate the start of output samples for successive iterations and REG_BCNT specifies the number of iterations. The REG_BCNT register is common for input formatter and output formatter. These registers can be better understood using the example given inFigure 10-10. Also, a complete use case is illustrated in Section 6 which provides further clarity on this aspect.

In the example shown in Figure 10-10, the output data consists of complex data (16-bit I and 16-bit Q) that is written to ACCEL_MEM3. The output data consists of four sets of 128 samples each (say, corresponding to FFT output of four RX antennas) and these are shown in four different colors. Each sample occupies 4 bytes and the samples are written to the output memory at a specific start address inside ACCEL_MEM3, as shown in Figure 10-10. The samples for the four RX antennas are written to the memory in an interleaved manner. Thus, for this example, a value of DSTADDR = 0xC040, DSTAINDX = 16, DSTACNT = 127, and DSTBINDX = 4 are used. The register REG_BCNT (common for input formatter and output formatter) is configured with a value of 3, corresponding to the four iterations required (for the four RX antennas). In steady state, for each clock cycle, the output formatter accepts one complex sample from the core computational unit and writes it into the memory as per the 2-D indexed addressing pattern programmed.

The register DSTACNT, which corresponds to the number of samples written to the destination memory for each iteration does not need to be equal to SRCACNT. This is useful in cases where some of the output samples (for example, some FFT bins at the end) can be dropped and do not need to be written into the destination memory. Another register, REG_DST_SKIP_INIT is also available, which can be used to skip some samples in the beginning as well. The number of samples written to the destination memory for each iteration is equal to (DSTACNT + 1) – REG_DST_SKIP_INIT.

Note that when performing FFT operations, internally the core computational unit sends out FFT output data in bit-reversed addressing order, but this is automatically handled in the output formatter, such that when the FFT output samples are written into the destination memory, they are written out in the correct normal order. Therefore, no special procedure is required on the part of the main processor to read the FFT output samples in the right sequence.

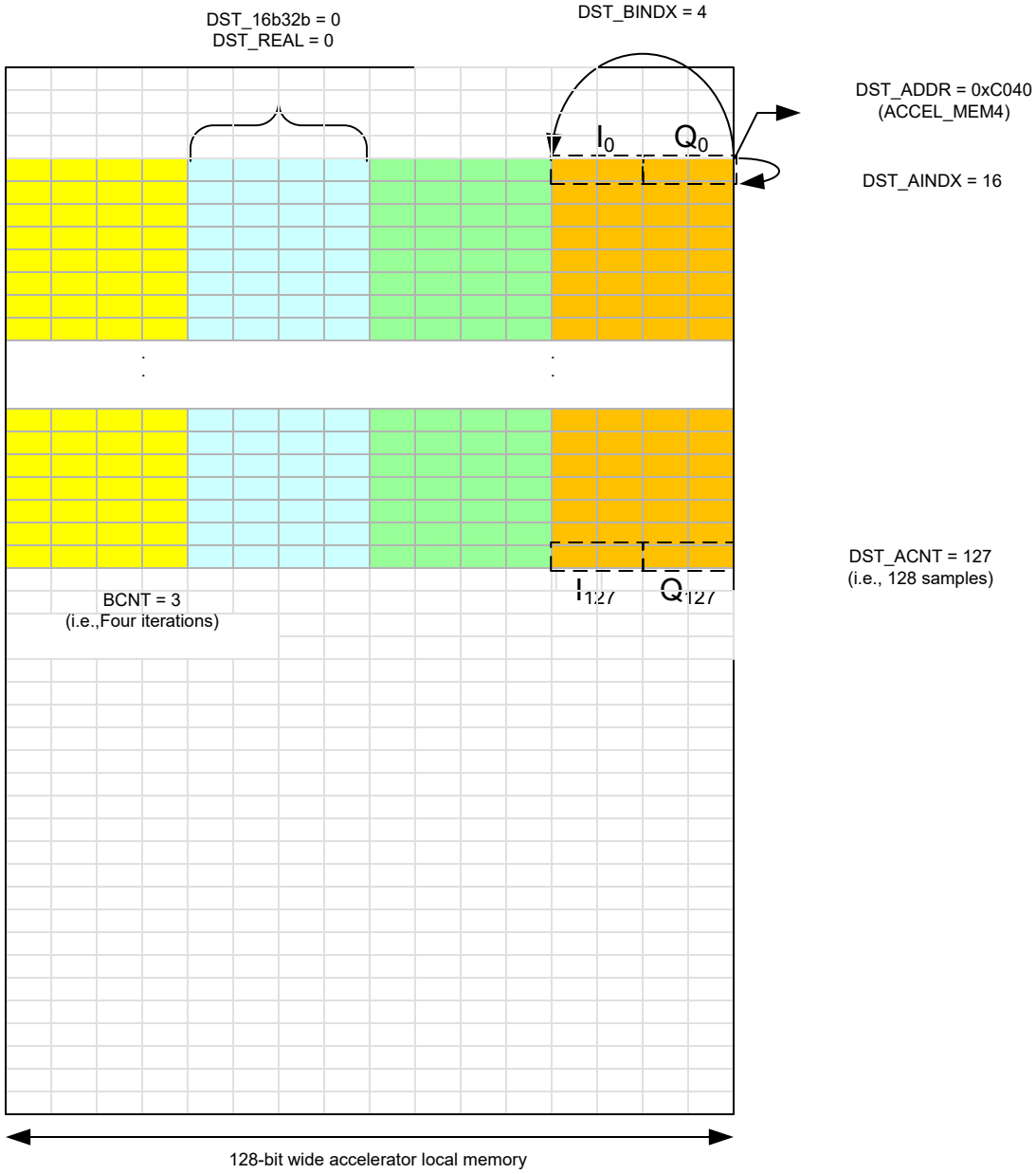


Figure 10-10. Output Formatter Destination Memory Access Pattern (Example)

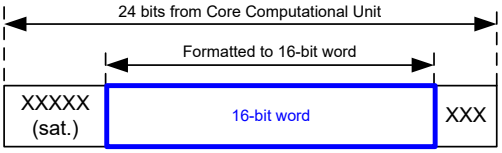
10.1.4.1.3 Output Formatter – Scaling and Formatting

The output formatter allows the 24-bit output samples from the core computational unit to be scaled and formatted before writing them to the destination memory as 16-bit or 32-bit words. There is a REG_DSTSCAL register which provides scaling options using bit-shift, to take the 24-bit samples and convert them to 16-bit or 32-bit data.

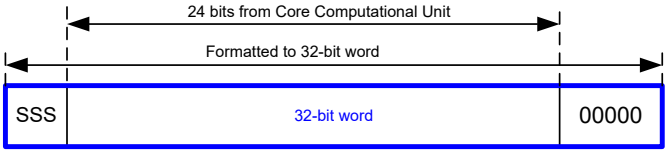
For the 16-bit case, the 24-bit sample (24-bits for each I and Q) is converted to 16-bit word by dropping REG_DSTSCAL bits at the LSB and by clipping with saturation (8-REG_DSTSCAL) bits at the MSB. For the 32-bit case, the 24-bit sample is padded with REG_DSTSCAL extra bits at the MSB and with (8- REG_DSTSCAL)

extra zeros at the LSB. Note that the register bit DSTSIGNED is used to indicate whether the output samples are signed or unsigned. When this register bit is set, the output samples are treated as signed numbers and therefore any extra MSB bits are sign-extended and any clipping of MSB bits handles signed saturation. In most cases of interest in part one of this user's guide (for example, when performing FFT operation), the output samples would be signed and therefore DSTSIGNED should be set (for example, equal to 1). However, if the log-magnitude operation in the core computational unit is enabled, then the output samples are unsigned and therefore DSTSIGNED is cleared (for example, equal to zero).

When the output samples are complex (for example, DSTREAL = 0), there is a provision to conjugate the output samples. Setting the register bit DSTCONJ conjugates the output samples before writing them to the destination memory. This feature (together with a corresponding SRCCONJ register bit in the input formatter block) enables an IFFT mode from the FFT engine.



For 16-bit case, if DST_SCAL = 3, then 3 bits are dropped at the LSB, and 5 bits are clipped (saturated) at the LSB



For 32-bit case, if DST_SCAL = 3, then 5 zeros are padded at the LSB, and 3 bits are extended at the MSB

Figure 10-11. Output Formatter Data Scaling

10.1.4.1.4 Output Formatter – Register Descriptions

Table 10-3 lists all the registers of the output formatter block.

Table 10-3. Output Formatter Registers

Register	Width	ParameterSet	Description
DSTADDR	16	Yes	Destination start address: This register specifies the starting address of the output samples, for example, it specifies the destination memory start address at which the output samples have to be written by the output formatter. This is a byte-address and this 16-bit register covers the entire address space of the four local memories (4 × 16KB = 64 KB). The four accelerator local memories are contiguous in the memory address space and any of them can act as the destination memory (as long as the same memory bank is not configured to be used as source memory at the same time).
DSTACNT	12	Yes	Destination sample count: This register specifies the number of samples (minus 1) to be written to the destination memory for every iteration. The sample count is in number of samples, not number of bytes. For example, the sample count can be specified as 191 (DSTACNT = 0x0BF) in a case where 192 samples must be written. Note that the DSTACNT register can be different from SRCACNT or even the FFT size. This is useful when only a part of the FFT bins must be written to memory and the remaining (far-end FFT bins) can be discarded. This register description is true when the REG_DST_SKIP_INIT register value is zero (see further for more information related to REG_DST_SKIP_INIT).

Table 10-3. Output Formatter Registers (continued)

Register	Width	ParameterSet	Description
DSTAINDX	16	Yes	Destination sample index increment: This register specifies the number of bytes separating successive samples to be written to the destination memory. For example, a value of DSTAINDX = 16 means that successive samples written to the destination memory should be separated by 16 bytes. The maximum value allowed for this register is 32767.
DSTBINDX	16	Yes	Destination offset per iteration: This register specifies the number of bytes separating the starting address of output samples for successive iterations. For example, when using four iterations to process four RX chains, this register can be used to specify the offset in the starting address between the successive RX chains. Note the distinction that DSTAINDX specifies the number of bytes separating successive samples for a particular iteration, whereas SRCBINDX specifies the number of bytes separating the starting address of the first sample for successive iterations. The maximum value allowed for this register is 32767.
REG_DST_SKIP_ INIT	10	Yes	Destination skip sample count: This register specifies how many output samples should be skipped in the beginning, before starting to write to the destination memory. This is useful if only a certain part of the FFT output (skipping the first several bins) need to be stored in memory. The total number of samples written to destination memory is equal to DSTACNT+1-REG_DST_SKIP_INIT.
DSTREAL	1	Yes	Complex or real output: This register-bit specifies whether the output samples are real or complex. A value of DSTREAL = 0 implies complex output and a value of DSTREAL = 1 implies real output. When real output is selected, the output formatter block automatically stores only the real part into the destination memory. This is useful when the core computational unit is configured to output magnitude or log-magnitude values.
DST16b32b	1	Yes	16-bit or 32-bit output word alignment: This register-bit specifies whether the output samples are to be written as 16-bits or 32- bits wide in the destination memory. A value of DST16b32b = 0 implies that the output samples are to be written as 16-bit words (in case of complex output, real and imaginary parts are each 16 bits wide). A value of DST16b32b = 1 implies that the output samples are 32-bits wide each.
DSTSIGNED	1	Yes	Output sign-extension mode: This register-bit, when set, specifies that the output samples are signed numbers and therefore, sign-extension or signed-saturation at the MSB is required when converting the 24-bit wide samples coming from the core computational unit into 16-bit or 32-bit output words to be written to the destination memory.

Table 10-3. Output Formatter Registers (continued)

Register	Width	ParameterSet	Description
DSTCONJ	1	Yes	Output conjugation: This register-bit specifies whether the output samples must be conjugated before writing them into the destination memory. If DSTCONJ is set, then the output samples are conjugated. Setting this register-bit only makes sense if the samples are complex numbers (for example, DSTREAL = 0). This register, together with its counterpart in the output formatter block, enables an IFFT mode for the FFT engine.
REG_DSTSCAL	4	Yes	Output scaling: This register specifies a programmable scaling using bit-shift, when converting the 24-bit samples coming from the core computational unit into 16-bit or 32-bit wide words to be written to the destination memory. See Figure 10-11 and itsdescription for more details regarding this register.
STATERRCODE (see Table 10-73.)	4	No	Memory access error: This 4-bit read-only register indicates if there is a memory access error caused by incorrect configuration or usage of the accelerator, where both the DMA and the accelerator are attempting to access the same 16KB memory at the same time. The 4- bit register indicates the error status for the 4 16KB memories (MSB bit corresponds to ACCEL_MEM0).
ERRCODEMASK (see Table 10-73.)	4	No	Mask for memory error: This register can be used to mask the memory access error. If set, the memory access error indication is disabled.
ERRCODECLR (see Table 10-73.)	4	No	Clear memory access error: This register can be used to clear the memory access error indication. Setting this register clears the error indication.

10.1.5 Accelerator Engine – Core Computational Unit

This section describes the core computational unit present in the accelerator engine (see Figure 10-12).

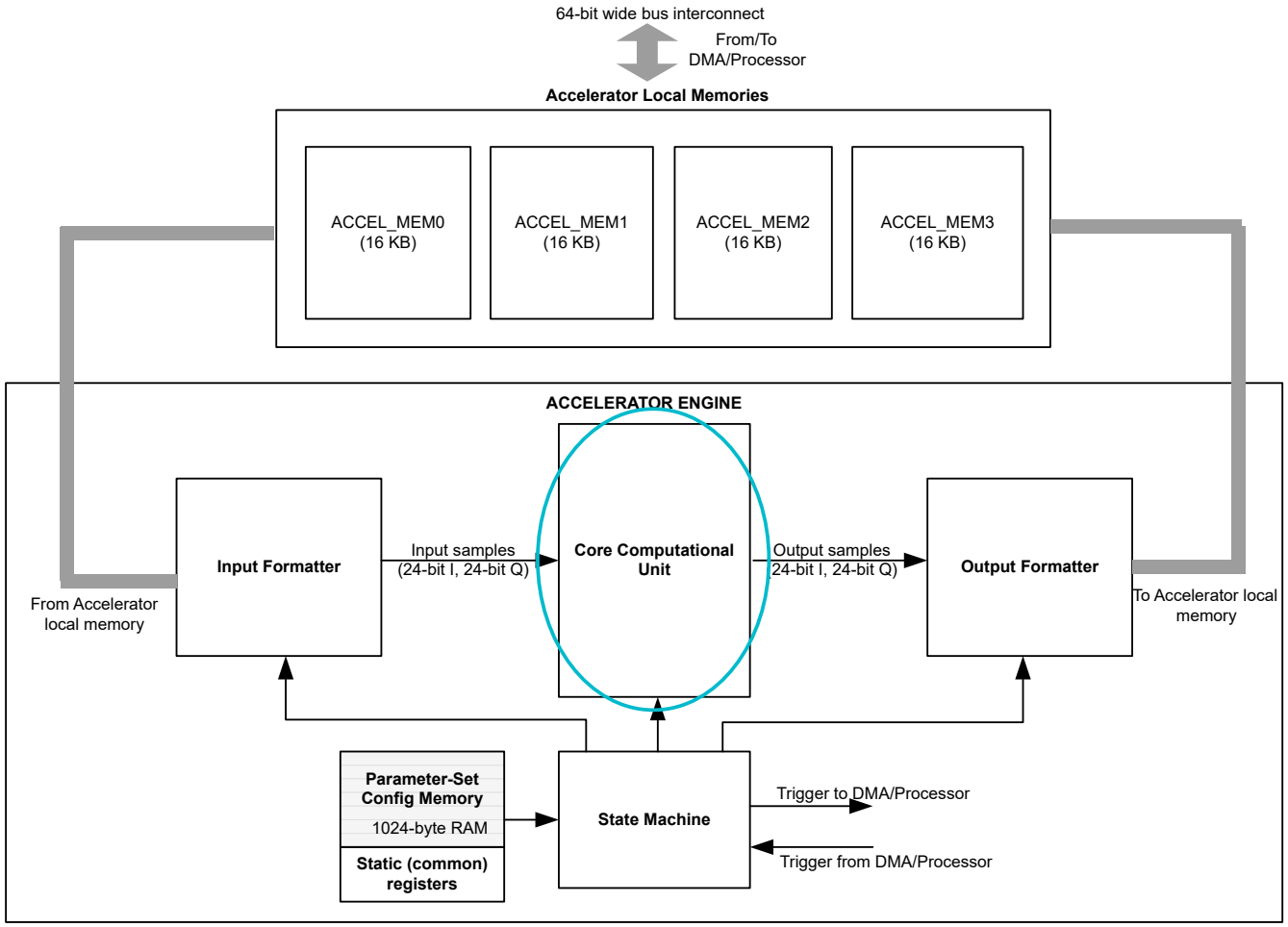


Figure 10-12. Core Computational Unit

10.1.5.1 Core Computational Unit

The core computational unit performs the mathematical operations required for the key functions, such as FFT, log-magnitude, and so on. The core computational unit accepts a streaming 24-bit complex input (24 bits for each I and Q) from the input formatter block and it outputs a streaming 24-bit complex output (24 bits for each I and Q) to the output formatter block. In addition to FFT and log-magnitude, the core computational unit has provision for simple pre-FFT processing, such as DC subtraction, zeroing out large interference samples, complex de-rotation, and windowing prior to FFT. The core computational unit also contains a CFAR-CA/OS detector unit for detecting peak samples (for example, radar targets).

Figure 10-13 shows the block diagram of the core computational unit. The core computational unit has three main paths – namely the FFT Engine path, CFAR Engine path and Compression Engine path. Only one of these three paths can be operational at any given instant. However, in separate parameter sets, different paths can be configured and used, so that multiple parameter sets executing one after another can accomplish a sequence of computational operations as desired. The register ACCEL_MODE controls which path gets used in a given parameter set.

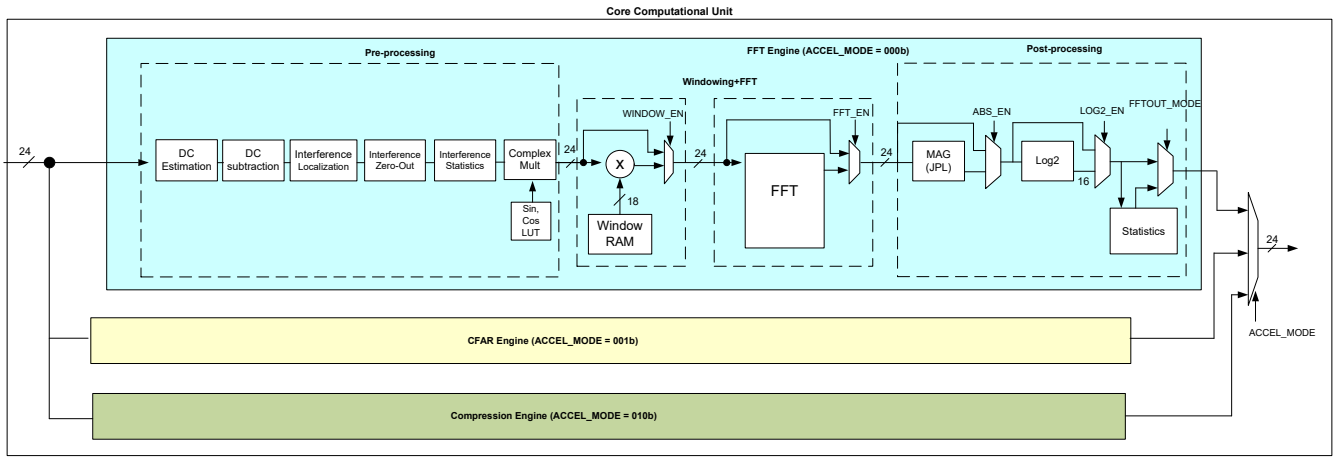


Figure 10-13. Core Computational Unit Block Diagram

For the purpose of part one of the user's guide, only the FFT Engine path is described. Specifically, the windowing, FFT, and log-magnitude operations are covered in this document. The other blocks in Figure 10-13, namely the Pre-processing, Statistics, CFAR Engine, Compression Engine are covered in part two of the user's guide and can be ignored for the present purpose.

10.1.5.1.1 Core Computation unit – Operation

The core computational unit operates on the streaming input of samples coming from the input formatter block, and in general outputs a stream of samples (after an initial latency in some cases) to the output formatter block. In general, at steady-state, one input sample is processed and one output sample is produced every 80-MHz clock.

The core computational unit has the ability to perform windowing, FFT, and log-magnitude computations. Each of these computational subblocks operate on a streaming input and produce a streaming output at the throughput of one sample per clock. These computational subblocks are stitched together one after the other in a series, as shown in Figure 10-13. This architecture allows multiple operations to be done in a streaming manner (for example, windowing and FFT can be done together), while at the same time, providing the user flexibility to choose one operation at a time.

The parameter-set registers WINDOW_EN, FFT_EN, ABSEN, and LOG2EN control the multiplexers (see Figure 10-13), which decide what operations are performed on the input samples for that parameter set.

Note that for the purpose of part one of the user's guide, the registers ACCEL_MODE and FFTOUT_MODE must be kept at zero. The purpose of these registers is covered in part two.

10.1.5.1.2 Core Computational Unit – Windowing

The incoming samples from the input formatter to the core computational unit are passed through the (optional) windowing operation (see Figure 10-13). Windowing operation is often required prior to performing FFT, to mitigate the sinc roll-off leakage from one strong FFT bin to the adjacent bins.

The implementation of the windowing operation in the accelerator is very straightforward. The window coefficients are preloaded by the Cortex-M4 processor into a dedicated Window RAM. The purpose of this RAM is to provide a fully programmable window (for example, Hann, Kaiser, or any proprietary window) to the user. Window RAM is single-bank memory each of 1K real 18-bit samples.

As the incoming samples from the input formatter stream in, each sample is multiplied by the appropriate window coefficient read from the RAM. Because the incoming samples are complex 24-bits wide (24-bits for each I and Q), the windowing operation involves multiplying the 24-bit I and 24-bit Q of the incoming sample with the 18-bit real window coefficient (see Figure 10-14). The output of this multiplication is rounded back to 24-bit I and 24-bit Q by dropping excess LSBs. Notethat windowing can be enabled or disabled by using the register bit WINDOW_EN.

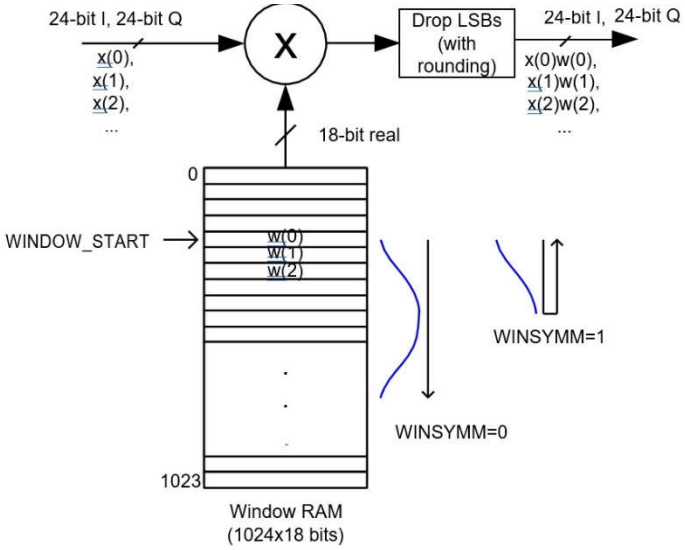


Figure 10-14. Windowing Computation

The start address (for example, starting coefficient index between 0 to 1023) is programmed in a 10-bit register WINDOW_START as part of the parameter set, so that the windowing computation can pick the appropriate window coefficients starting from that index. For each incoming sample, the index keeps incrementing, so that each successive sample is multiplied by the successive window coefficient. At the end of each iteration (for example, when SRCACNT number of samples have been processed), the index resets back to the starting coefficient index programmed for the parameter set, so that the next iteration can be performed. At the end of all the iterations of the current parameter set, the next parameter set can use a different window if desired. For example, when performing second- and third-dimension FFTs one after another (in two parameter sets), the window functions for both these FFTs can be prestored in the Window RAM and appropriate start index can be provided for each of the FFT operation dimensions.

If the window function is symmetric, only one half of the set of window coefficients needs must be stored in the Window RAM. The register bit WINSYMM, when set, indicates that after SRCACNT / 2 samples (or, if SRCACNT is odd, (SRCACNT + 1) / 2 samples) are processed, the window coefficients read-indexing must be reversed, so that the same set of coefficients used for the first SRCACNT / 2 samples are reused in the reverse order for the next SRCACNT / 2 samples. (See Figure 10-14). If SRCACNT is even, then the last window coefficient is read only once, when the direction is reversed. If SRCACNT is odd, then the last window coefficients is read twice, when the direction is reversed.

The output of the windowing computation is 24-bit I and 24-bit Q, which is streamed into the FFT subblock.

10.1.5.1.3 Core Computational Unit – FFT

The FFT subblock performs FFT on the incoming 24-bit I and 24-bit Q data stream. The FFT sizes supported are all powers of 2 until 1024, for example, FFT sizes of 2, 4, 8, 16, … 512 and 1024 are supported. The lowest FFT size of 2 is mostly useful as a complex add-subtract feature or while using the FFT stitching feature. FFT sizes of 4, 8, 16, and 32 can be used for third dimension (angle estimation) FFT.

Note that FFT stitching is a feature that enables large FFT sizes, specifically, 2048 and 4096, using a two- step process (this feature is not covered here and is discussed in part two of the user's guide).

The FFT operation can be enabled or disabled by using the register bit FFT_EN. When enabled, the FFT subblock computes the FFT of the input data stream and produces a 24-bit I and 24-bit Q output stream. This output stream is initially in bit-reversed order, but the output formatter handles appropriately writing the output to the destination memory in the correct order.

The FFT implementation comprises ten butterfly stages. Depending on the FFT size needed, an appropriate number of butterfly stages are employed. The FFT size is programmed using the FFTSIZE register – for example, FFTSIZE = 5 means 32-point FFT, FFTSIZE = 7 means 128-point FFT, and so on. Note that the FFT size must be equal to or larger than SRCACNT, and the input formatter block automatically zero-pads extra

samples to account for the difference between FFT size and SRCACNT. For example, if SRCACNT = 99 (for example, 100 samples) and FFTSIZE = 7 (for example, 128-point FFT), then the input formatter automatically appends 28 zero-pad samples for each iteration.

10.1.5.1.4 Core Computational Unit – FFT Quantization and Speed performance

As is well known, a butterfly stage typically consists of add-subtract and twiddle multiplication operations. At the output of each add-subtract structure, the bit-width would increase by 1 bit (for example, 24-bit input would grow to 25-bit output). To handle this one-bit growth due to add-subtract operation, there is a provision at the output of each butterfly add-subtract stage to scale the result back to 24 bits, by either dividing the output by 2 (round off one LSB) or by saturating one MSB, shown in Figure 10-15.

The 10-bit register BFLY_SCALING is used to control this divide-by-2 scaling operation at each stage, so that the user has full flexibility to control the signal level through the different butterfly stages. If BFLY_SCALING = 0 for a particular stage, then the 25-bit output is saturated at the MSB to get back to 24 bits. Otherwise, it is convergent-rounded at the LSB to get back to 24 bits. The user can thus control the scaling at each of the ten butterfly stages. The LSB of this 10-bit register corresponds to the last stage and the MSB of this register corresponds to the first stage. For an FFT size of 64, only the LSB 6 bits are relevant.

There is a 10-bit read-only register FFTCLIP which indicates whether there was any clipping in any of the butterfly stages. This register is a sticky register that gets set when a clipping event occurs and remains set until it is cleared using the CLR_FFTCLIP register bit. See the register description of FFTCLIP in Table 5.

The twiddle factors are stored as 24-bit I and 24-bit Q coefficients. Prior to twiddle factor multiplication, the coefficients are reduced to 21-bit I and 21-bit Q by dropping three LSBs (with optional dithering). The purpose of dithering is to eliminate any repetitive quantization noise patterns from degrading the SFDR of the FFT. The use of dithering here is optional. For dithering, an LFSR is used to generate a random pattern, for which the LFSR seed must be loaded with a non-zero value (see LFSRSEED in the register descriptions).

The SFDR performance of the FFT, with dithering enabled, is better than –140 dBc, as shown in Figure 10-16. The architecture of the FFT is such that it can take a streaming input (one sample per clock) and produce a streaming FFT output (one sample per clock), in steady-state. There is an initial latency of approximately FFT size number of clocks. This latency only comes into picture once for a given parameter set. Within a parameter set, multiple FFT iterations can be performed back-to-back (for example, for four RX) with no additional latency between iterations. Because the implementation uses 80-MHz clock, a 256-point complex FFT for four RX chains would take 256 + 256 × 4 clock cycles to complete, which corresponds to 16 µs (plus a few clocks of implementation latencies, which are not accounted here). Table 4 lists the approximate computation time needed for various FFT sizes. The output of the FFT can be fed to the output formatter or it can be sent to the magnitude/log-magnitude computation subblock.

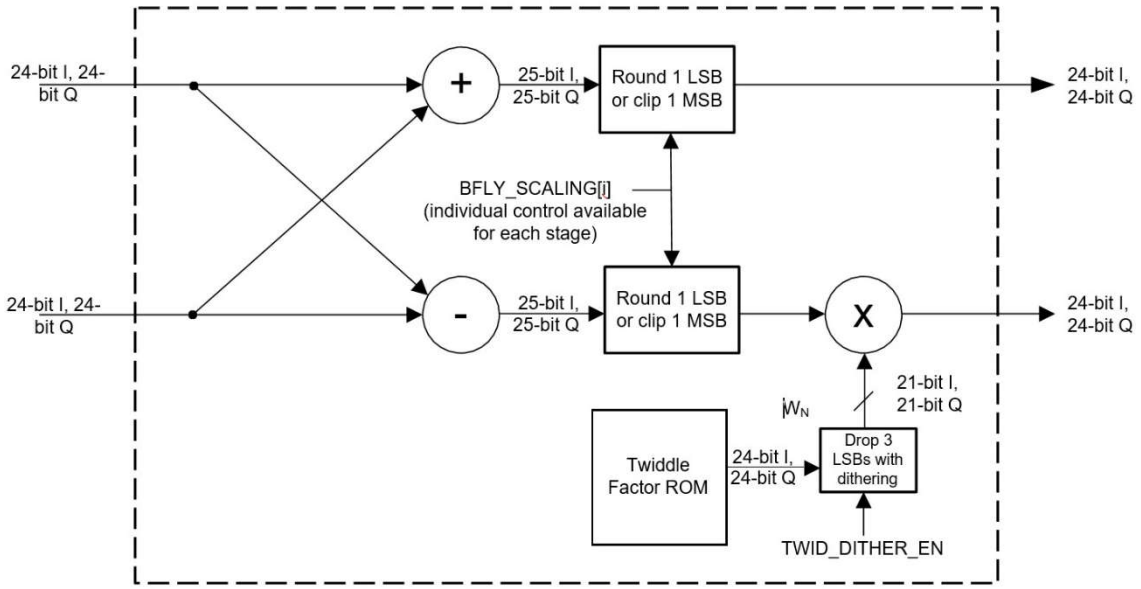


Figure 10-15. ButterflyStage Fixed-Point

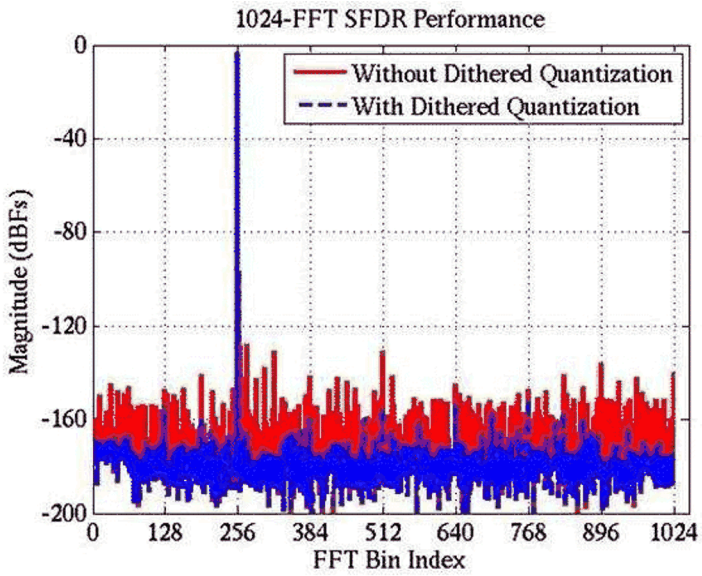


Figure 10-16. FFT SFDR Performance with andWithout Dithering

Table 10-4. FFT Computation Time

Example	FFT Size	Numberof Back-to-Back Iterations	Numberof Clock Cycles (Initial latency + Computation)	Total Duration
1	256	4	256+ (256 × 4)	16µs
2	128	4	128+ (128 × 4)	8µs
3	8	64	8+ (64 × 8)	6.5µs

Note

The FFT is a complex FFT implementation. If the input samples are real-only, then the SRCREAL register bit can be set, such that the imaginary part (Q-part) will be forced to zero by the input formatter block.

ADVANCE INFORMATION

10.1.5.1.5 Core Computational Unit – Magnitude and Log-Magnitude Post-Processing

The magnitude and log-magnitude post-processing block computes absolute value or log2 of the absolute value of its input. Because this block is connected to the output of the FFT engine, the computation of absolute value (and log2) can be directly performed on the streaming FFT output. Alternately, the FFT block can be bypassed and only the magnitude and log-magnitude block can be employed.

The processing in this block first involves computation of magnitude (absolute value) of the input samples in the magnitude subblock (using 3-segment linear approximation). The result of the magnitude computation is fed into a Log2 computation subblock, which uses a look-up table-based approximation to compute logarithm- base-2 of the magnitude.

As shown in Figure 10-13, if the register-bit ABSEN is set, the magnitude computation subblock is enabled. In addition, if the register-bit LOG2EN is set, then the Log2 computation subblock is also enabled. Note that setting LOG2EN makes sense only when ABSEN is also set. The magnitude output is 24-bits wide (real number).

Next,the log2 computation of the magnitude value is achieved as follows. Any unsigned input number N can be written as N = 2^k(1 + f) and the log2(N) can then be written as follows in the equation below:

$$\log2(N) = k + \log2(1+f). \quad (1) \qquad (1)$$

The implementation of log2 computation uses the previous formula, where a look-up table approximation is used to generate the second term, for example, log2(1 + f). The log2 output is 16-bits wide. The 16-bit logarithm output consists of 5 bits of integer part and 11 bits of fractional part.

The accuracy of magnitude and log-magnitude are shown in #none#

Magnitude error (dB)	Log-Mag error (dB)
0.051	0.0055

Depending on the settings of ABS_EN and LOG2_EN, either the magnitude or the log-magnitude is sent as the final output of the core computational unit. The final output of the core computational unit going to the output formatter is 24-bits I and 24-bits Q. Thus, if either magnitude or log-magnitude is enabled, the Q- values are just made zeros. Similarly, when log2 is enabled, because the output is 16-bits, 8 MSBs are filled as zero.

The output formatter handles writing the samples to the destination memory as per the configured destination memory access pattern described in a previous section.

10.1.5.1.6 Core Computational Unit – Register Descriptions

lists all the registers of the core computational unit.

Table 10-5.

Register	Width	ParameterSet	Description
WINDOW_EN	1	Yes	Windowing Enable: This register-bitenables or disables the pre-FFT windowing operation. If this register is set to 1, then the windowing is enabled, otherwise, it is disabled. The exact window function (coefficients) to be applied is specified in a dedicated Window RAM, which is 1024 × 18 bits in size.
FFT_EN	1	Yes	FFT Enable: This register-bitis used to enable the FFT computation. If FFT_EN = 1, then the FFT computation is enabled. Otherwise, it is disabled (bypassed).

Table 10-5. (continued)

Register	Width	ParameterSet	Description
ABS_EN	1	Yes	MagnitudeEnable: This register-bitis used to enable the magnitude calculation. If this register bit is set, then the magnitude calculation is enabled, else it is bypassed. When enabled, the magnitude (absolute value) of the input complex samples are calculated using JPL approximation and the resulting magnitude value is sent on the I-arm of the output. The Q-arm is made zeros.
LOG2_EN	1	Yes	Log2Enable: This register-bitis used to enable the Log2 computation. If this register bit is set, then the Log2 computation is enabled, else it is bypassed. Note that setting this register bit only makes sense if the inputs to the Log2 computation are unsigned real numbers, such as when the Magnitude Enable bit (ABSEN) is also set. When enabled, the Log2 of the magnitude of the input samples is calculated and sent out on the I-arm of the output. The Q-arm is made zeros.
WINDOW_START	10	Yes	Windowing coefficientsstart index: This registerspecifies the starting index of the window coefficients within the Window RAM. The value of this register ranges from 0 to 1023. The purpose of this register is to allow multiple windows (for example, one window of 512 coefficients and another window of 256 coefficients) to be stored in the Window RAM and one of these windows can be used by programming this start index register appropriately in the current parameter set.
WINSYMM	1	Yes	Window symmetry: This register-bitindicates whether the complete set of window coefficients are stored in the Window RAM or whether one half of the coefficients are stored. If this register bit is set, it means that the window function is symmetric and therefore, only one half of the window function coefficients are stored in the Window RAM. See the description section related to Windowing computation for more details.

Table 10-5. (continued)

Register	Width	ParameterSet	Description
FFTSIZE	4	Yes	FFT size: This registerspecifies the FFT size. The mapping of the FFTSIZE register to the actual FFT size is as follows: Actual FFT size = 2 ^ (FFTSIZE). For example, a register value of 0110b specifies that the FFT size is 64. The maximum FFT size that is supported is 1024. Therefore, this register value is never expected to exceed 1010b. Note that the FFT size should be equal to or larger than SRCACNT and the Input Formatter block will automatically zero-pad extra samples to account for the difference between FFT size and SRCACNT. For large-size FFT (> 1024 point) that might be useful for industrial level-sensing applications, an FFT stitching procedure is supported, which is based on performing multiple smaller size FFTs in a first step and then stitching them in a second step (using a subsequent parameter set). This FFT stitching feature is covered in part two of the user's guide.
BFLY_SCALING	10	Yes	Butterfly scaling: This registeris used to control the butterfly scaling at each stage of the FFT structure. Because the maximum FFT size is 1024, there are up to ten butterfly stages. Each butterfly stage has an add-and-subtract structure, at the output of which the bit-width would temporarily increase by 1 (from 24 to 25 bits wide). If BFLY_SCALING = 0, then the 25-bit output is saturated at the MSB to get back to 24 bits. Otherwise, it is convergent-rounded at the LSB to get back to 24 bits. The user can thus control the scaling at each of the 10 butterfly stages. The LSB of this register corresponds to the last stage and the MSB of this register corresponds to the first stage. For an FFT size of 64, only the LSB 6 bits are relevant.
DITHERTWIDEN (see Table 10-23.)	1	No	Twiddle factor ditheringenable: This register-bitis used to enable and disable dithering of twiddle factors in the FFT. The twiddle factors are 24-bits wide (24-bits for each I and Q), but they are quantized to 21-bits before twiddle factor multiplication. This quantization is implemented with dithering on the LSB, to avoid periodic quantization pattern affecting SFDR performance of the FFT. This is optional.
LFSRSEED (see Table 10-25.)	29	No	Seed for LFSR(random pattern): For twiddle factor dithering,there is an LFSR that is used, whose seed value is loaded by writing to this 29-bit LFSRSEED register. The LFSRSEED register should be set to any non-zero value, say 0x1234567.

Table 10-5. (continued)

Register	Width	ParameterSet	Description
LFSRLOAD (see Table 10-25.)	1	No	To loadthe LFSR seed, a pulse signal needs to be provided, by writing a 1 followed by a 0 (i.e., by setting and clearing) the LFSRLOAD register-bit.
FFTCLIPSTAT (see Table 10-74.)	10	No	FFT ClipStatus (read-only): This isa read-only status register, which indicates any saturation/clipping events that have happened in the FFT butterfly stages. Note that each of the 10 butterfly stages in the FFT can be programmed to either saturate the MSB or round the LSB. Whenever saturation of MSB is used in any stage, there is a possibility that that stage can saturate or clip samples. In that case, this saturation event is indicated in the corresponding bit in this status register, so that the Cortex-M4F processor can read it. If multiple FFTs are performed, this status register includes any saturation events happening in any of them. This status register can only be cleared by the M4F, by setting another single-bit register CLR_FFTCLIP, so that the saturation status indication gets cleared back to 0 and any subsequent saturation events can be freshly monitored.
CLRFFTCLIPSTAT (see Table 10-74.)	1	No	ClearFFT Clip Status register: This registerbit, when set, clears the FFTCLIP register.
ACCEL_MODE	3	Yes	Select CoreComputational Unit Data Path: This registerselects the data-path mode of the accelerator's core computational unit – for example, it selects whether the FFT engine path the CFAR engine path or compression engine path is active. This register will be covered in part two and part 3. For the purpose of part one of the user's guide, this register should be zero. 000 – FFT 001 – CFAR 010 – Compression 111 – NOP

10.1.6 Appendix

PARAM-set Layouts for three engines are shown below

Parameter-set for FFT Engine path (REG_ACCEL_MODE = 000b)																																
S.No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0	DCEST_RESET_MODE(2)		INTF_RESET_MODE (2)	INTF_THRESH_SEL (2)	FFT_OUTPUT_MODE (2)	ACCEL_MODE (3)	CMULT_MODE (3)	ABSEN (1)	LOG2EN (1)	WINDOW_EN (1)	FFT_EN (1)	BPM_EN (1)	ACC2DMA_CHANNEL_TRIGDST (4)	DMATRIGEN (1)	CM4INTREN (1)	DMA2ACC_CHANNEL_TRIGSRC (4)	TRIGMODE (3)	
1	DSTADDR (16)									SRCADDR (16)								
2	DSTCONJ (1)	DSTSIGNED (1)	DST16b32b (1)	DSTREAL (1)	DSTACNT (12)					SRCCONJ (1)	SRCSIGNED (1)	SRC16b32b (1)	SRCREAL (1)	SRCACNT (12)				
3	DSTAINDX (16)									SRCAINDX (16)								
4	DSTBINDX (16)									SRCBINDX (16)								
5	DCSUB_EN(1)	DCSUB_SEL(1)	REG_DST_SKIP_INIT (10)				REG_DSTSCAL (4)			REG_SRCSCAL (4)			REG_BCNT (12)					
6	BFLY_SCALING (10)					WINDOW_START (10)				BPMPHASE (4)	INTF_THRESH_EN (1)	WINSYMM (1)	FFTSIZE (4)	INTF_ZERO_MODE(2)				
7	CIRCSHIFTWRAP (4)		WINDOW_INTERP_FRACTION (2)	TWIDINCR (14)						CIRCIRSHIFT (12)								

Parameter-set for CFAR Engine path (REG_ACCEL_MODE = 001b)																																
S.No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	RESERVED (8)							ACCEL_MODE (3)		CFAR_CYCLIC (1)		CFAR_INP_MODE (1)		CFAR_LOG_MODE (1)		CFAR_ABS_MODE (2)		CFAR_GROUPING_EN (1)		RESERVED (2)		ACC2DMA_CHANNEL_TRIGDST (4)		DMATRIGEN (1)		CR4INTREN (1)	DMA2ACC_CHANNEL_TRIGSRC (4)					TRIGMODE (3)	
1	DSTADDR (16)																SRCADDR (16)															
2	DSTCONJ (1)	DSTSIGNED (1)	DST16b32b (1)	DSTREAL (1)	DSTACNT (12)												SRCCONJ (1)	SRCSIGNED (1)	SRC16b32b (1)	SRCREAL (1)	SRCACNT (12)											
3	DSTAINDX (16)																SRCAINDX (16)															
4	DSTBINDX (16)																SRCBINDX (16)															
5	RESERVED (2)		REG_DST_SKIP_INIT (10)									REG_DSTSCAL (4)					REG_SRCSCAL (4)				REG_BCNT (12)											

6	RESERVED (13)	CFAR_NOISE_DIV (4)	CFAR_GUARD_INT (3)	RESERVED (5)	CFAR_OS_NON_CYC_VARIANT_EN(1)	CFAR_OS_KVALUE(6)		
7	CIRCSHIFTWRAP (4)	CFAR_OUT_MODE (2)	CFAR_AVG_LEFT (6)	CFAR_AVG_RIGHT (6)	CFAR_CA_MODE (2)	CIRCIRSHIFT (12)		

Parameter-set for CMP-DCMP Engine path (REG_ACCEL_MODE = 010b)																														
S.No.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	RESERVED (8)								ACCEL_MODE (3) = 010b	RESERVED (8)										ACC2DMA_TRIGDST (4)				DMATRIGEN (1)	INTREN (1)	DMA2ACC_TRIGSRC (4)				TRIGMODE (3)			
1	DSTADDR (16)																SRCADDR (16)														
2	DSTCONJ (1)	DSTSIGNED (1)	DST16b32b (1)	DSTREAL (1)	DSTACNT (12)												SRCCONJ (1)	SRCSIGNED (1)	SRC16b32b (1)	SRCREAL (1)	SRCACNT (12)											
3	DSTAINDX (16)																SRCAINDX (16)														
4	DSTBINDX (16)																SRCBINDX (16)														
5	RESERVED (2)	DST_SKIP_INIT (10)										DSTSCAL (4)					SRCSCAL (4)			BCNT (12)											
6	CMP_SCALEFAC (5)				CMP_EGE_OPT_K_INDX (4)				CMP_PASS_SEL(2)		CMP_HEADER_EN (1)	CMP_SCALEFAC_BW(4)	CMP_BFP_MANTISSA_BW (5)							CMP_EGE_K_ARR_LEN (4)				CMP_METHOD (3)		CMP_DCMP (1)		CMP_DITHER_EN(1)		RESERVED(2)		
7	RESERVED (32)																															

10.2 Radar Hardware Accelerator - Part 2

This part of the HWA user guide is organized as follows:

- Section 1 covers some additional features of the core computational unit related to pre-FFT processing.
- Section 2 covers details of CFAR-CA & CFAR-OS detection features.
- Section 3 covers other miscellaneous capabilities such as statistics computation.

10.2.1 FFT Engine - Pre-processing

As explained in Part 1 of the user guide, the FFT Engine comprises pre-processing, windowing, FFT and Log-magnitude subblocks and these are stitched together one after the other in series (refer Figure 10-17). This architecture allows multiple operations to be done in a streaming manner (for example, windowing and FFT can be done together), while at the same time, providing the user flexibility to choose one operation at a time. This section provides an overview of the pre-processing subblock inside the FFT engine of the core computational unit.

ADVANCE INFORMATION

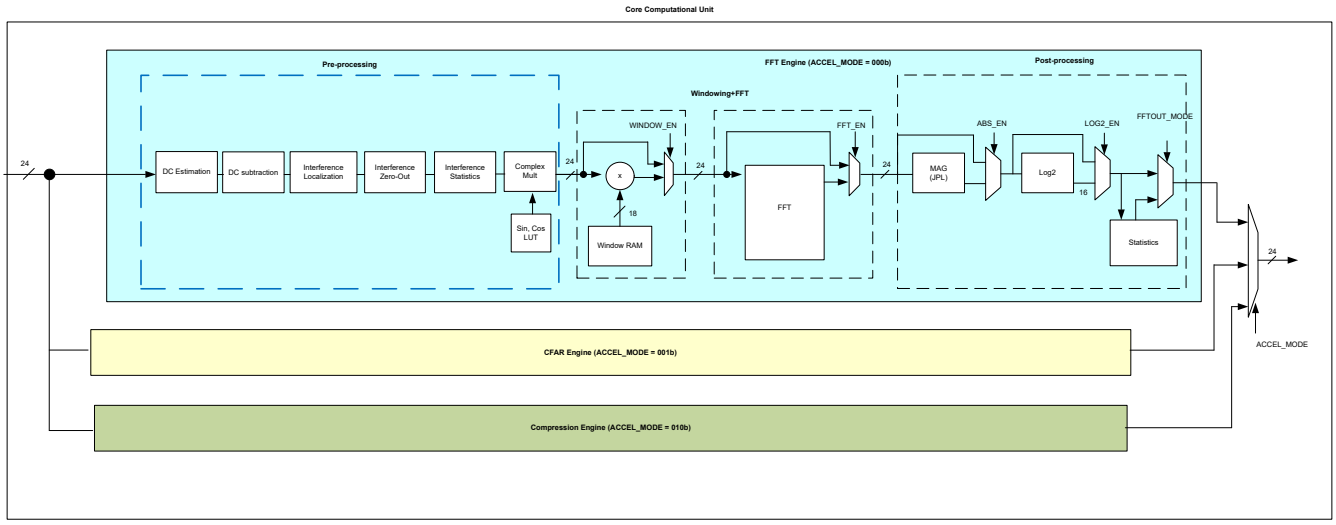


Figure 10-17. FFT Pre-processing

The pre-processing subblock also provides capability for DC estimation and correction, Interference localization and zero-out, interference statistics and complex multiplication.

10.2.1.1 DC Correction

10.2.1.1.1 DC Estimation

The DC estimation block estimates the time-domain average of the stream of samples along the A dimension. The stream can be one chirp, or set of chirps, i.e., frame. The DC is estimated on a per-iteration basis (i.e., along A dimension for each B iteration) for I & Q samples. Up-to 6 estimates corresponding up to 6 independent input streams are available.

DC estimation is based on accumulation followed by a fine scaling and a programmable right shift. The fine scaling is configured as 1.8 value, via the 9-bit DCEST_SCALE register. The subsequent programmable right shift is configurable from 2 to 17. Therefore, the DC estimation is well suited for cases where the number of samples per iteration is between 2^2 and 2^17. The fixed-point details are captured in Figure 10-18. The internal accumulator reset supports several modes as shown in Table 1. For example, when DCEST_RESET_MODE = 2, the internal DC accumulators are reset at the beginning of the current parameter-set execution. Therefore, this mode estimates DC value for each set of SRCACNT samples along the A-dimension for up to 6 iterations along B-dimension within the current parameter-set. This mode is useful for per-chirp DC estimation. In this mode, the estimated DC values per iteration are latched at the end of current param-set. On the other hand, when DCEST_RESET_MODE = 3, the internal DC accumulators are reset only when the state machine executes the first loop of the parameter-set. As the state machine loops through various parameter-sets multiple times as programmed via NLOOPS register, the DC accumulators are not reset in between these loops. This mode is useful for per-frame DC estimation, where each loop corresponds to one chirp and the NLOOPS loops (chirps) correspond to a complete frame. The estimated DC values per iteration are latched at the end of last execution of the param-set.

The processor can read the DC estimates through the read-only registers – DCESTI_0VAL, .., DCESTI_5VAL & DCESTQ_0VAL, .., DCESTQ_5VAL. The DC estimates can also be used for DC subtraction described next.

Table 10-6. DC Estimation – Reset modes

DCEST_RESET_MODE	Comments
0	Hold the DC internal accumulators without updating (bypass DC estimation).
1	DC estimation enabled, but free-running without automatic reset (i.e., not reset at the start of this parameter-set). In this mode, the software can reset the DC accumulators by writing to DC_EST_RESET_SW register bit.
2	Reset the DC internal accumulators at the start of this parameter-set. This mode is applicable for per-chirp DC estimation.

Table 10-6. DC Estimation – Reset modes (continued)

3	Reset the DC internal accumulators at the start of this parameter-set only if the loop-counter is 0. This mode is applicable for per-frame DC estimation.

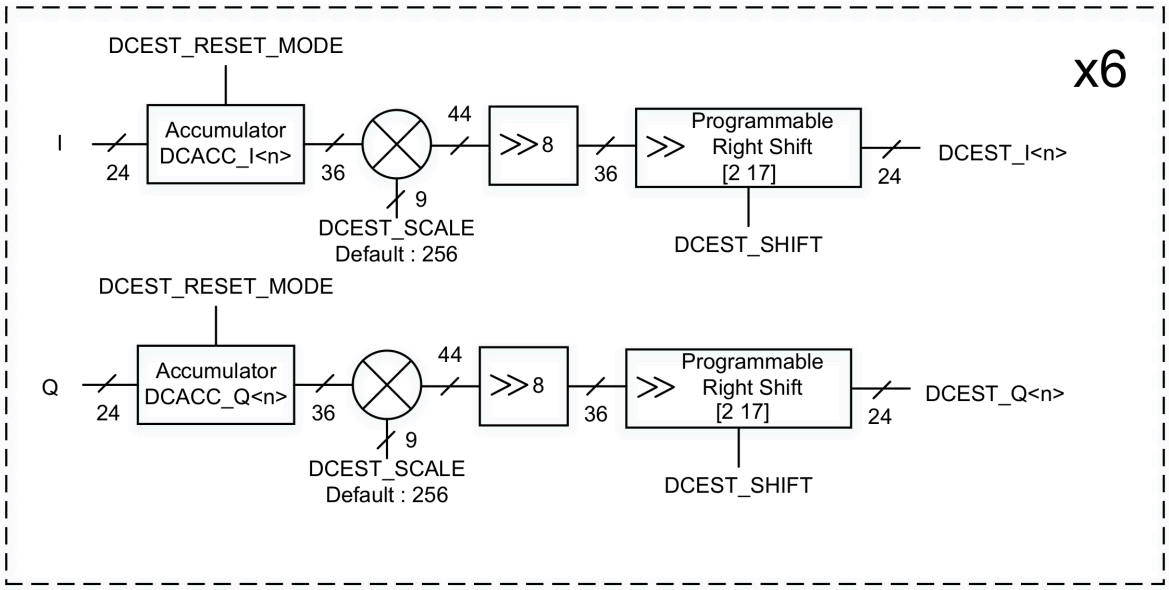


Figure 10-18. DC estimation

10.2.1.1.2 DC Subtraction

The DC subtraction feature is enabled if the register DCSUB_EN is set to 1.

DC subtraction (see Figure 10-19) can use the output from the built-in DC estimation accumulators, or a user-programmed value, based on the register bit, DCSUB_SELECT. If DCSUB_SELECT is 1, the DC estimation based on the internal accumulators is used. If DCSUB_SELECT is 0, the software override values are used (they are given by DC_SW_I_<n> and DC_SW_Q_<n> for the nth iteration).

When using the built-in DC estimation accumulators, DC subtraction is performed on 6 individual streams corresponding to say 6 channels on a per-iteration basis. Note that in a typical usage, for performing per-chirp DC estimation and DC subtraction, a two-pass approach is needed, where the first pass is configured for DC estimation via one parameter-set, and the second pass is configured for DC subtraction in the next parameter-set. Alternately, if a previous DC estimate (eg. From the previous frame) is desired to be used for DC subtraction for the current chirp, then DC subtraction can be directly accomplished in one pass.

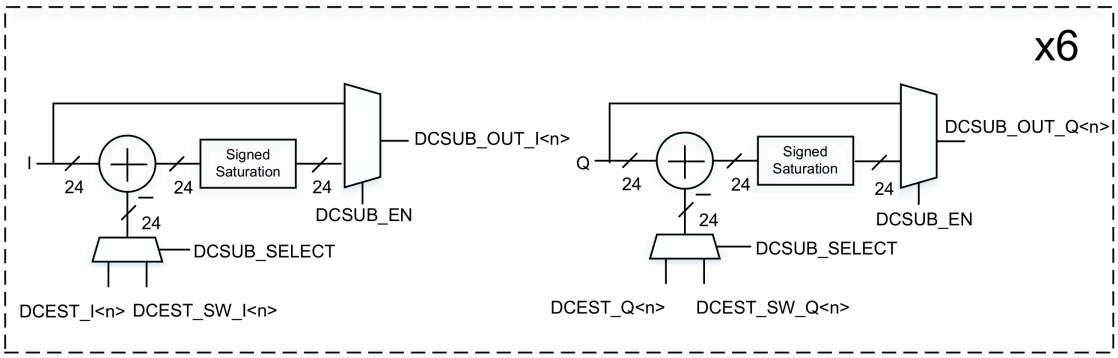


Figure 10-19. DC Subtraction

10.2.1.2 Interference Mitigation

10.2.1.2.1 Interference Localization and Zero-out

In an FMCW radar transceiver, interference from another radar typically manifests itself as a time-domain spike in a few samples. This spike corresponds to the time duration when the chirping frequency of both radars overlap with each other. Such a time-domain spike caused by interference can lead to degradation in the noise floor at the FFT output, causing degradation in detection performance.

In order to mitigate the impact of interference, the pre-processing block provides capability to perform interference localization to identify samples corrupted by interference, followed by interference zero-out to repair those samples.

The INTF_THRESH_EN register is provided as part of the parameter-set to control when the interference localization should be enabled. When enabled, the input samples are fed through a magnitude calculation (based on JPL approximation), which computes a 24-bit magnitude of the 24-bit input complex sample. For definition of this approximation, see Part 1 of this user guide. Similarly, magnitude of the backward difference between adjacent samples is also computed, which is another useful metric for interference (glitch) detection.

Any sample whose magnitude and/or magnitude of backward difference exceeds thresholds THRESH_MAG<n> and THRESH_MAGDIFF<n> is considered as affected by interference and is zeroed-out This is supported individually for up to 6 iterations. The register, INTF_ZERO_MODE determines the logic to combine the thresholds using the magnitude and/or magnitude of difference estimates. Based on this register, samples are zeroed-out if they exceed the THRESH_MAG<n>, or THRESH_MAGDIFF<n>, a logical AND of both thresholds, a logical OR of both as shown in Figure 10-20. This applies across all iterations

As shown in Figure 10-21, The threshold values of THRESH_MAG<n> and THRESH_MAGDIFF<n> applied on a per-channel basis can derived from user-programmed registers – INTF_THRESH_MAG<n>_SW, INTF_THRESH_MAGDIFF<n>_SW or from taken from built-in Interference statistics block – INTF_THRESH_MAG<n>, INTF_THRESH_MAGDIFF<n> as described in the next section. The user can also choose to sum the built-in interference statistics estimates across all channels to derive a common interference threshold across all iterations– INTF_STATS_SUM_MAG, INTF_STATS_SUM_MAGDIFF. The register, INTF_THRESH_SELECT is used to select among these threshold options.

The number of samples marked with IIB across the iterations is recorded in the read-only registers, INTF_COUNT_ALL_CHIRP and INTF_COUNT_ALL_FRAME. This can be read after every chirp or after the completion of a frame (when the state machine completes all the programmed parameter-set loops and enters idle state).

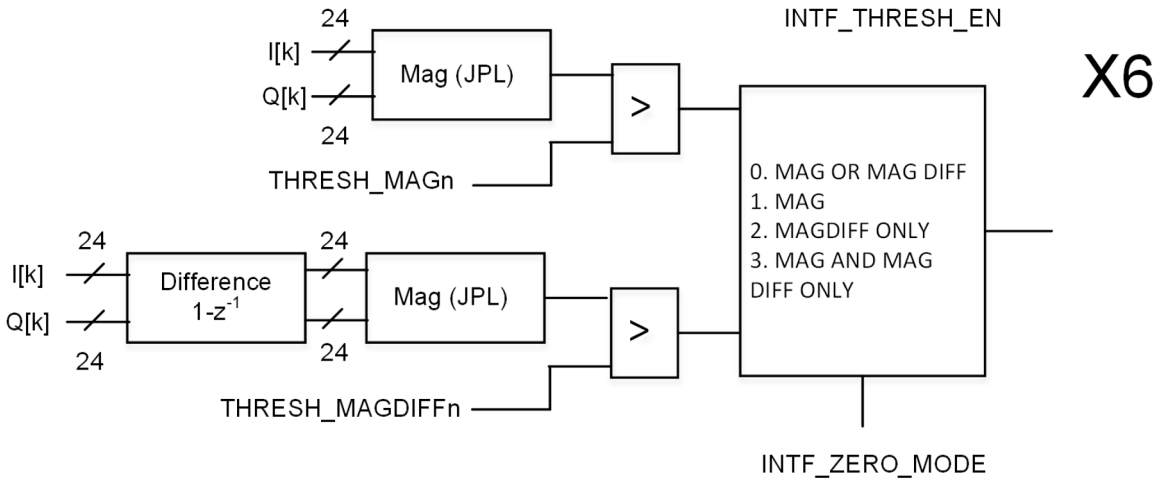


Figure 10-20. Interference Localization and Zero-out

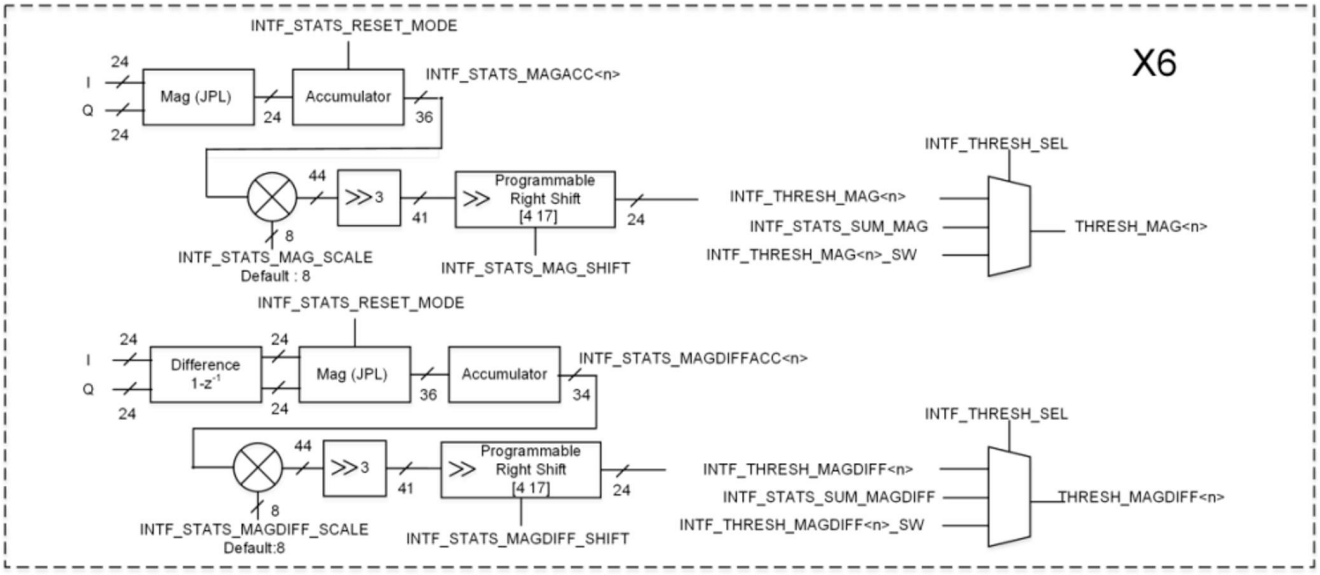


Figure 10-21. Interference Statistics

10.2.1.2.2 Interference Statistics

Figure 10-21 provides the thresholds for interference localization. In order to obtain the interference statistics and derive the thresholds, the magnitude and magnitude of backward difference of the incoming samples are accumulated per iteration and up to 6 such independent accumulations are supported. These registers can be reset on a per-chirp or per-frame basis, and the behavior can be controlled using the register INTF_STATS_RESET_MODE. These reset modes are similar to DCEST_RESET_MODE previously explained (refer Table 1). The interference statistics accumulators can be reset by software via writing the INTF_STATS_RESET_SW register bit. This reset also clears the INTF _COUNT_ALLs

The determination of interference threshold for interference localization is based on taking the above accumulator values and applying a programmable fine scaling, followed by a programmable right shift. The fine scaling is configured via 8-bit registers, INTF_STATS_MAG_SCALE and INTF_STATS_MAGDIFF_SCALE in 5.3 format. The fine scaling value is interpreted as an unsigned 8-bit number with 5 integer bits and 3 fractional bits giving a scale in range [0 to 31.875]. The default value of this register is 8, applying a scaling of 1.0. The programmable right-shift in the range of 4 to 17 is applied via the registers, INTF_STATS_MAG_SHIFT and INTF_STATS_MAGDIFF_SHIFT respectively. Note that if the sum mode of threshold selection is made, then the shift values have to include the extra division based on number of iterations being summed. The resulting values INTF_THRESH_MAGn and INTF_THRESH_MAGDIFFn are used as thresholds in the interference localization block as described in the previous section.

10.2.1.3 Complex Multiplication

In addition to interference zero-out, the pre-processing block contains a complex multiplication sub-block. The purpose of this sub-block (Figure 10-22) is to enable several assorted capabilities that require complex multiplication of the input samples. The CMULT_MODE register is used to enable and configure the complex multiplication functionality. The complex multiplication sub-block can be disabled (bypassed) by the setting CMULT_MODE to 0b0000. Any other value of this register will enable the complex multiplication sub-block and configure it to perform specific operation as described in the next few paragraphs.

There are seven modes of the complex multiplier supported as follows. They are frequency shifter mode, frequency shifter with auto-increment mode (a slow DFT mode), FFT stitching mode, magnitude squared mode, scalar multiplication mode, vector multiplication modes-1 & 2. In all the seven modes of the complex multiplier, one complex multiplication is performed every clock cycle.

- Frequency shifter mode: If the register value is CMULT_MODE = 001b, then the complex multiplier functions as a frequency shifter, which can be used to de-rotate the input samples by a certain frequency. This de-rotation is accomplished using cos, sin values from a twiddle factor look-up table (LUT). This LUT contains the (compressed) equivalent of the cos, sin values corresponding to the 16384 long sequence

exp(-j*2*pi*(0:16383)/16384). Another register (TWIDINCR) is used to specify the de-rotation frequency, by specifying how much the phase should change for each successive input sample (that register controls how much the LUT read index increments every sample). In effect, the input samples x(n) for n = 0 to SRCACNT-1 are multiplied by the sequence, exp(-j*2*pi*TWIDINCR*(0:SRCACNT-1)/16384).

- Frequency shifter with auto-increment mode (a slow DFT mode): If the register value is CMULT_MODE = 010b, then the complex multiplier functions in a mode which enables Discrete Fourier Transform (DFT) computation. In this case, the complex multiplier performs a function that is very similar to frequency shifter mode, except that, at the end of each iteration, the de-rotation frequency is automatically incremented for the next iteration. Note that DFT computation for a given set of input samples involves de-rotating the samples by one frequency at a time, and computing a sum of the de-rotated samples for each such frequency. To achieve DFT computation, the Input Formatter should be configured to send the same set of input samples to the complex multiplier for multiple iterations (as many as the number of DFT bins required) and the complex multiplier de-rotates the samples by one frequency at a time and auto-increments to the next frequency for the next iteration. Also, the statistics block (explained in a later section) is used to compute the sum of the de-rotated samples corresponding to each iteration, which then becomes the final DFT value. The DFT computation is 'slow' in the sense that in each 80 MHz clock cycle, only one complex multiplication is performed. For example, for a 512-point input sample set, it would take 512 clock cycles per DFT bin. However, since the DFT mode is typically only used for FFT peak interpolation (very few bins), it is acceptable. The starting frequency for the DFT computation is specified in the TWIDINCR register (similar to the frequency shifter mode). The increment value by which the frequency increments every iteration is obtained from FFTSIZE register – Note that the DFT mode cannot be used simultaneously with FFT enabled, hence the FFTSIZE register has been over-loaded for providing the increment value in this mode. The increment value is calculated as 2^(14 – FFTSIZE) and hence the DFT resolution is 16384/2^(14 – FFTSIZE) = 2^FFTSIZE. As an example, if FFTSIZE = 1011b, then the DFT resolution is 2048. This is equivalent to computing DFT points corresponding to 2K size FFT grid. The highest resolution for the DFT would be obtained when FFTSIZE = 1110b (max allowed value), in which case the DFT resolution is 16384 (corresponding to 16K size FFT grid). In effect, for the kth iteration (with k starting from 0), the input samples x(n) for n = 0 to SRCACNT-1 are multiplied by the sequence, exp(j*2*pi*(TWIDINCR+2^(14-FFTSIZE)*k)*(0:SRCACNT-1)/16384).
- FFT Stitching mode: If the register value is CMULT_MODE = 011b, then the complex multiplier functions in FFT stitching mode. This mode is useful when large size FFTs (2K and 4K) are required. Since the FFT block natively supports only up to 1024 size, for 2048 and 4096 point FFT, an FFT Stitching procedure using two steps (two parameter-sets) can be used. As an example, when a 4K size FFT is needed, it is achieved in two steps as follows. In the first step, every 4th input sample is passed through a 1K size FFT (four 1K point FFTs are performed on decimated input samples). Then, in the next step, the resulting 4x1024 FFT outputs are sent through four-point "stitching" FFTs (1024 four point FFTs), with an additional pre-multiplication by the complex multiplier block to achieve FFT stitching. This pre-multiplication uses the twiddle factor LUT in a specific pattern, for which additional configuration information is available in TWIDINCR register (2 LSB bits). If the LSB two bits of TWIDINCR register are 00b, then the twiddle factor pattern will correspond to what is required for 2K (2x1024) size FFT stitching. If the LSB two bits are 01b, then the twiddle factor pattern will correspond to what is required for 4K (4x1024) size FFT stitching. Values of 10b and 11b are reserved and should not be used. Also, the unused 12 MSB bits of TWIDINCR register must be zero in this mode of operation.
- Magnitude squared mode: If the register value is CULT_MODE = 100b, then the complex multiplier functions in magnitude squared mode. In this case, the complex multiplier takes every complex input and produce the magnitude squared as the output. This can be used together with the statistics block (explained in Section 3) to compute the mean squared sum of the input samples.
- Scalar multiplication mode: If the register value is CMULT_MODE = 101b, then the complex Multiplier functions in scalar multiplication mode. This feature is useful if the input samples need to be scaled by some constant factor. In this case, the complex multiplier will multiply each input sample with a 21-bit scalar complex number that is programmed in ICMULTSCALE<n> and QCMULTSCALE<n> registers (for I and Q value, each having 21 bits). The ICMULTSCALE<n> and QCMULTSCALE<n> registers are common registers and not part of parameter-set. Note that this feature can be used to multiply the input samples for different iterations (channels) with different complex scalars for up-to 6 iterations.
- Vector multiplication mode 1: If the register value is CMULT_MODE = 110b, then the complex multiplier functions in vector multiplication mode 1. The purpose of this mode is to enable element-wise multiplication

of two complex vectors, as well as dot-product capability (using statistics block to sum the element-wise multiplication output). The samples from the Input Formatter block constitute one of the two vectors, whereas the other vector is taken from a pre-loaded internal RAM inside the core computational unit. (This internal RAM is a RAM that is normally used by the FFT block when performing 1024-point FFT computation). This internal RAM can store 512-complex samples and hence the vector multiplication can support a maximum of 512 elements of multiplication. The Vector multiplication is not a highly parallelized operation, in the sense that only one complex multiplication is done per 80 MHz clock cycle.

It is important to note one important limitation: since the internal RAM is shared with the FFT,performing a 1024-point FFT destroys the pre-loaded contents of the RAM. Therefore, performingvector multiplication and 1024-point FFT back-to-back many times requires re-loading of the internalRAM each time and will be inefficient. However, note that this limitation of having to re-load the internalRAM does not apply when performing FFT of size 512 or less, which is often the case for second andthird dimension FFTs.

The operation of the vector multiplication mode 1 is as follows. The streaming set samples from the Input Formatter block coming at 80 MHz is element-wise multiplied with successive samples from the internal RAM. The statistics block (described in a later section) can be used to compute the sum for every iteration, which enables a dot-product implementation if desired. At the end of every iteration, the addressing from the internal RAM is reset, so that for the next iteration, the samples are picked up from the start of the internal RAM.

- Vector multiplication mode 2: If the register value is CMULT_MODE = 111b, then the complexmultiplier functions in vector multiplication mode 2, which is slightly different from the earlier mode. Theonly difference in this case is that at the end of every iteration, the addressing of the internal RAM isnot reset, so that for the next iteration, the samples from the internal RAM are picked up with anaddress that continues from where it left off at the end of the previous iteration. This mode can be usedwhen a given set of input samples needs to be element-wise multiplied with multiple vectors. In thiscase, the input formatter block can be configured to repeat the same set of samples for multipleiterations, and the internal RAM can be loaded with all the vectors, such that for successive iterations,the input samples are multiplied with successive vectors.

For loading the internal RAM used for the vector multiplication modes, the register bit STG1LUTSELWR isused. The internal RAM for vector multiplication mode is mapped to the same address space as theWindow RAM. Therefore, this register bit is required to specify which of these two (Window RAM orinternal RAM) need to be selected, when loading the co-efficients via DMA or main processor. If theregister bit is 0, then the Window RAM is selected, else, the internal RAM for vector multiplication mode isselected. Note that the other registers such as WINDOW_START, which pertains to windowing, arealways applicable only for the Window RAM. The STG1LUTSELWR register bit should in general be keptas 0 (Window RAM selected). This allows the main processor or DMA to have access to the Window RAMby default. Only when it is desired to load the internal RAM with coefficients for vector multiplication mode,this register bit should be temporarily set to 1. After loading the coefficients, the register bit should bemade 0 again. Note that in all the above seven modes of the complex multiplier, only one complex multiplication is performed every 80 MHz clock. So, the effective speed achieved for the multiply or multiply-accumulate operation is 80 MHz.

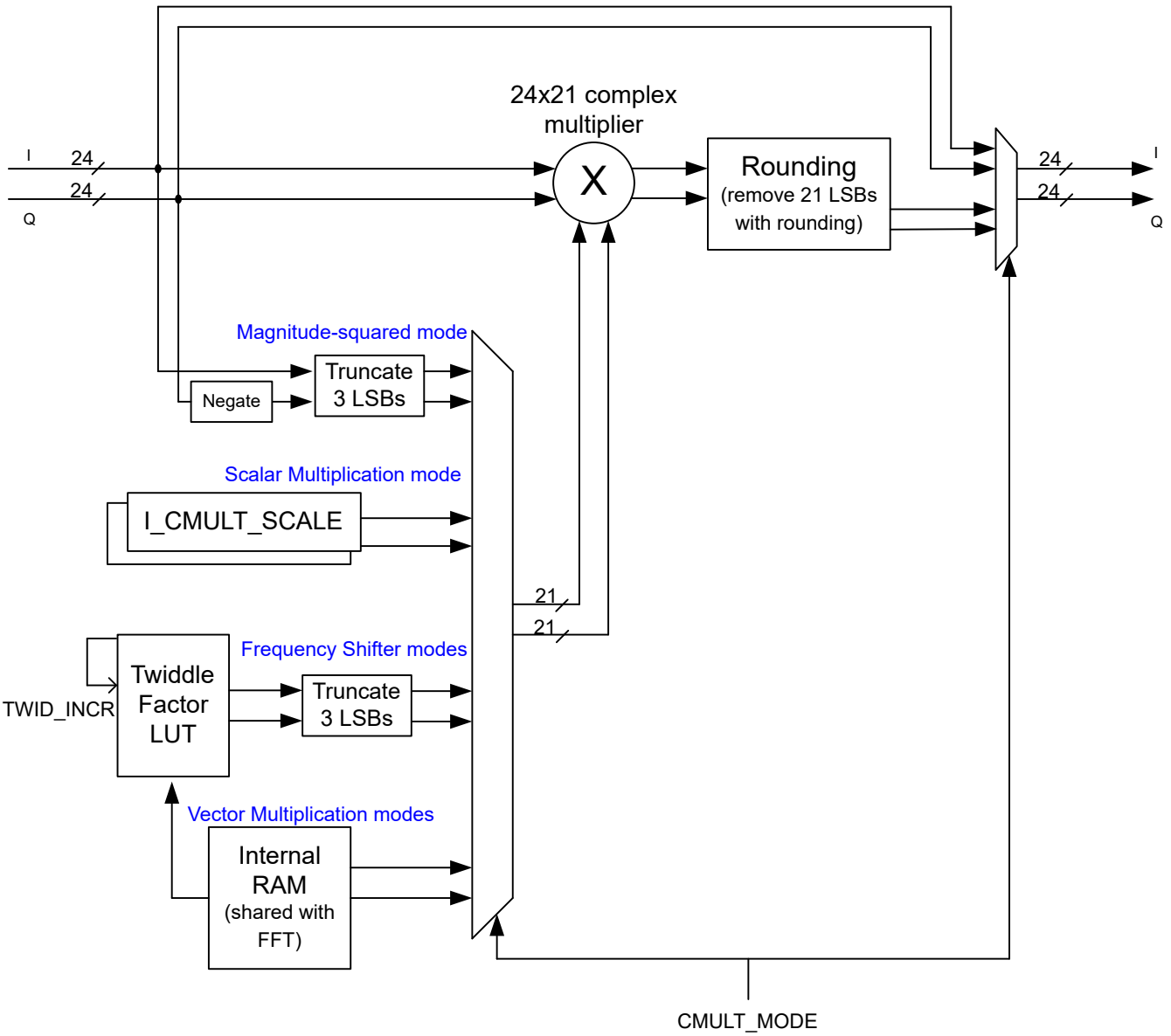


Figure 10-22. Complex multiplication capability in Pre-processing block

10.2.1.4 BPM Removal

Although not explicitly shown in Figure 10-17, it is possible to multiply the input samples going from the inputformatter into the core computational unit with a +1/-1 programmable binary sequence (of length up to 64).This feature is enabled by setting the register bit BPM_EN in the parameter-set.This feature may be useful when Binary Phase Modulation (BPM) is used during transmission of chirps.The BPM pattern is generally a pseudo-random sequence (chipping sequence) of 1's and -1's, which havealready been applied to the radar transmit signal. Therefore, the radar signal processing of the resultantanalog-to-digital converter (ADC) samples prior to FFT needs to undo the modulation. For instance, ifeach chirp is transmitted with a +1 or -1 polarity, then it is necessary to undo this sequence prior to the second dimension FFT processing across chirps. The BPM removal feature can be used to achieve this.

When BPM removal is enabled, each input sample is multiplied by a +1 or -1, based on the sequencepresent in the 64-bit BPMPATTERNLSB and BPMPATTERNMSB register. The register BPMRATE is usedto control for how many consecutive samples the same BPM bit is applied. For example, if BPMRATE = 4,then the same BPM bit is applied for 4 consecutive samples. Similarly, if BPMRATE = 1, then the BPM bitis changed for every sample.

There is another register BPMPHASE that specifies the number of consecutive samples for which the firstBPM bit is applied. Note that this is applicable only for the first BPM bit. If BPMPHASE = 0, then the firstBPM bit is applied for BPMRATE number of samples. Otherwise, the first BPM bit is applied forBPMRATE – BPMPHASE number of samples. For example, if BPMPHASE = 1 and BPMRATE = 4, thenthe first BPM bit is applied for 4-1 = 3 samples, and then subsequent BPM bits are applied with periodicityof 4 samples for each bit. This is shown in Figure 10-23.

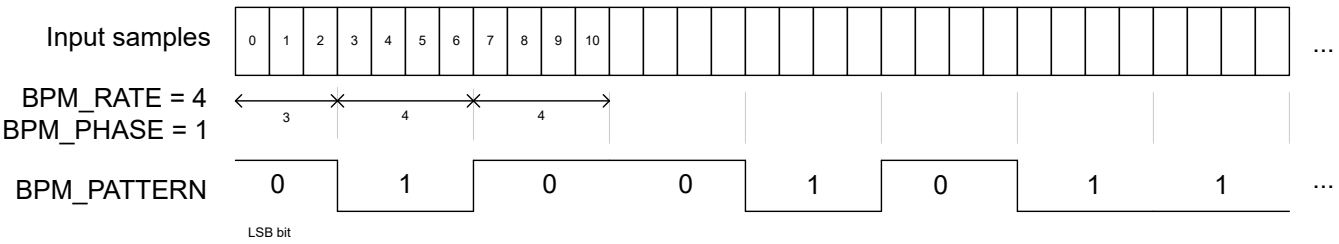


Figure 10-23. BPM Removal Capability

If multiple iterations (for example, four back-to-back FFTs in a single parameter-set using REG_BCNT=3)are done, then the same BPM pattern gets applied to the input samples in each iteration.Note the limitation that the BPM pattern register is 64 bits long, hence, the maximum BPM sequenceLength that is supported is 64. For higher BPM sequence length, the alternate approach of pre-multiplyingthe window coefficients stored in the window RAM may be considered.

10.2.1.5 Pre-Processing Block Register Descriptions

Table 10-7. Pre-Processing Block Registers

Register.field	Width	Parameter-Set? (Y/N)	Description
DCEST_RESET_MODE	2	Y	2-bit field that controls the reset behavior for all 12 DC accumulators 00 : Hold Accumulator state without updating 01 : Reserved 10 : Reset at start of param-set (i.e., per-chirp DC estimation). 11 : Reset at start of param-set only if loop counter is 0 (i.e., per-frame DC estimation)
DC_EST_SCALE	9	N	Programmable fine scaling for DC estimation: 9-bit scale applied to all 12 DC accumulators. This is followed by right shift and truncation. Multiplies the accumulator output by DC_EST_SCALE/256. Default value is 256 giving a scale of 1.0. Setting it to 128, gives a scale of 0.5.
DC_EST_SHIFT	4	N	Programmable right shift for DC estimation: Right bit-shift applied to all 6 DC accumulator outputs. Cannot be bypassed. Accumulator outputs are scaled by 2^(8 + 2 + DC_EST_SHIFT). Valid range for this register is 0 to 15 (i.e., scaling of 2^2 to 2^17). Note that DC_EST_SHIFT > 15 is not supported.
DC_ACC_I_<n>_VAL_LSB n=0,1,.5	32	N	These read-only registers provide the lower 32 bits of 36b DC estimation accumulator values –I&Q for 6 streams for processor read-out.
DC_ACC_I_<n>_VAL_MSB n=0,1,..5	4	N	These read-only registers provide the upper 4 bits of 36b DC estimation accumulator values –I&Q for 6 streams for processor read-out.
DC_EST_RESET_SW	1	N	Software reset for DC accumulators: Setting this register bit to 1 resets all 6 DC estimation accumulators. This is a self-clearing reset bit.

Table 10-7. Pre-Processing Block Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
DC_EST_I_<n>_VAL, DC_EST_Q_<n>_VAL n=0,1,..5	24	N	These read-only registers provide the DC estimates – I&Q for 6 streams – for the processor to read.
DC_ACC_CLIP_STATUS	6	N	Clip status indication (read-register) for the 12 DC accumulators (both I and Q combined). Value of 1 indicates a clipping event occurred.
DC_EST_CLIP_STATUS	6	N	Clip status indication (read-register) for the 12 DC estimates (both I and Q combined). Value of 1 indicates a clipping event occurred.
DCSUB_EN	1	Y	Enable or Disable DC subtraction. If this register bit is set to 1, DC subtraction is enabled. Else, it is disabled.
DCSUB_SELECT	1	Y	Source select for DC subtraction: 0 : Value comes from processor via DC_SW_I<n> & DC_SW_Q<n> 1: Value comes from built-in DC estimation hardware, i.e., DCEST_I<n> & DCEST_Q<n>
DC_I<n>_SW DC_Q<n>_SW n=0,1,..5	24	N	User-programmed DC values used for DC subtraction. These registers are relevant only when DCSUB_SELECT is 0.
DC_SUB_CLIP	1	N	Clip status indication (read-register) for DC subtraction node (both I and Q combined). Value of 1 indicates a clipping event occurred.
INTF_THRESH_EN	1	Y	Enable/Disable for Interference zeroing-out This registerbit controls the enable/disable for the interference zero-ing feature. The feature is enabled if this register bit is set to 1.
INTF_THRESH_MAG<n>_SW n=0..5	24	N	Software Interference threshold for Magnitude These registersare used to specify the user-programmed threshold for nulling out samples affected by interference in the Interference localization block. The magnitude of each incoming samples is compared with this threshold to decide whether it is corrupted by interference or not.
INTF_THRESH_MAGDIFF<n>_SW n=0…5	24	N	Software Interference threshold for Magnitude of backward difference These registersare used to specify the user-programmed threshold for nulling samples affected by interference in the Interference localization block. The magnitude of backward difference of incoming samples is compared with this threshold to decide whether it is corrupted by interference or not.
INTF_THRESH_MODE	2	Y	Interference detection mode selection: This register is used to control the mode for interference detection in the Interference localization block. 00 : Magnitude OR Magnitude difference 01: Only Magnitude difference 10: Only Magnitude 11 : Magnitude AND Magnitude difference

Table 10-7. Pre-Processing Block Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
INTF_THRESH_SEL	2	Y	Select the source of interference threshold 0 : User-defined threshold via INTERFTHRESH_MAG_SW and INTERFTHRESH_MAGDIFF_SW 1 : Single threshold based on built-in interference statistics outputs using sum value across collected interference statistics 2 : Threshold based on built-in interference statistics outputs, with each statistic being used for corresponding iteration (RX channel)
INTF_STATS_RESET_MODE	2	Y	Reset mode control for Interference statistics accumulators: Controls the reset behavior for all 12 magnitude and magdiff accumulators. 00 : Hold Accumulator state without updating 01 : Free-running accumulator mode 10 : Reset at start of parameter-set (i.e.,per-chirp accumulation). 11 : Reset at start of parameter-set only if loop counter is 0 (i.e., per-frame)
INTF_STATS_MAG_SCALE	8	N	Programmable fine scaling for Interference statistics Magnitude: Scaling applied to INTF_STATS_MAGACC<n> from interference statistics block.
INTF_STATS_MAG_SHIFT	3	N	Programmable right shift for Interference statistics Magnitude: Right bit-shift applied to the interference magnitude accumulator. Total right shift of the accumulator is 2^(3+4+INTF_STATS_MAG_SHIFT). Valid range for this register is 0 to 13 (i.e., the total right shift can't be more than 2^17).
INTF_STATS_MAGDIFF_SCALE	8	N	Programmable fine scaling for Interference statistics MagDiff: Scaling applied to INTF_STATS_MAGDIFFACC<n> from interference statistics block.
INTF_STATS_MAGDIFF_SHIFT	3	N	Programmable right shift for Interference statistics MagDiff: Right bit-shift applied to the interference magdiff accumulator. Total right shift of the accumulator is 2^(3+4+INTF_STATS_MAGDIFF_SHIFT). Valid range for this register is 0 to 13 (i.e., the right shift can't be more than 2^17).
INTF_STATS_MAG_ACC_<n>_LSB	32	N	These read-only registers provide the lower 32 bits of 36b magnitude accumulator values –I&Q 6 streams for processor read-out.
INTF_STATS_MAG_ACC_<n>_MSB	4	N	These read-only registers provide the upper 4 bits of 36b magnitude accumulator values –I&Q 6 streams for processor read-out.
INTF_STATS_MAGDIFF_ACC_<n>_LSB	32	N	These read-only registers provide the lower 32 bits of 36b magnitude difference accumulator values –I&Q 6 streams for processor read-out.
INTF_STATS_MAG_ACC_<n>_MSB	4	N	These read-only registers provide the upper 4 bits of 36b magnitude difference accumulator values –I&Q 6 streams for processor read-out.
INTF_STATS_RESET_SW	1	N	Software reset bit for all the interference statistics accumulators. This is a self-clearing reset bit.
INTF_THRESH_MAG<n>_VAL	24	N	Read-only thresholds – scaled and shifted INTF_STATS_MAGACC<n> of interference statistics block
INTF_STATS_SUM_MAG_VAL	24	N	Sum of INTF_LOC_THRESH_MAG<n>_VAL, based on number of iterations. Useful as single magnitude threshold value across all iterations

Table 10-7. Pre-Processing Block Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
INTF_THRESH_MAGDIFF<n>_VAL	24	N	Read-only thresholds – scaled and shifted INTF_STATS_MAGACCDIFF<n> of interference statistics block
INTF_STATS_SUM_MAGDIFF_VAL	24	N	Sum of INTF_LOC_THRESH_MAGDIFF<n>_VAL, based on number of iterations. Useful as single magnitude difference threshold value across all iterations
INTF_STATS_SUM_MAG_VAL_CLIP_STATUS	1	N	Read-only clip status for sum of all magnitude thresholds computed by the statistics block. Value of 1 indicates that the sum clipped.
INTF_STATS_SUM_MAGDIFF_VAL_CLIP_STATUS	1	N	Read-only clip status for sum of all magnitude difference thresholds computed by statistics block. Value of 1 indicates that the sum clipped.
INTF_STATS_ACC_CLIP_STATUS	6	N	Read-only clip status indication register for 6 magnitude based interference threshold. INTF_THRESH_MAG<n>_VAL.
INTF_STATS_ACC_CLIP_STATUS	6	N	Read-only clip status indication register for magnitude-difference based interference threshold. INTF_THRESH_MAGDIFF<n>_VAL.
INTF_STATS_THRESH_CLIP_STATUS.	6	N	Read-only clip status indication register for 6 interference statistics magnitude accumulators INTF_STATS_MAGACC<n>.
INTF_STATS_THRESH_CLIP_STATUS	6	N	Read-only clip status indication register for 6 interference statistics magnitude-difference accumulators INTF_STATS_MAGACCDIFF<n>.
INTF_LOC_COUNT_ALL_CHIRP	12	N	Read-only register indicating the number of samples that exceeded the threshold in a given param-set. The count is saturated to 2^12 – 1.
INTF_LOC_COUNT_ALL_FRAME	20	N	Read-only register indicating the number of samples that exceeded the threshold across multiple executions of same param-set. The count is saturated to 2^20 – 1.
CMULT_MODE	4	Y	Complexmultiplication mode selection: This registeris used to configure the mode of the complex multiplication sub-block. A value of 0000b disables/bypasses the complex multiplication. Any other value chooses one of nine available modes of operation . Detailed description of the nine modes in the main description section.
ICMULT_SCALE<n> QCMULT_SCALE<n>. n = 0 .5	21	N	Coefficients for Complex multiplication: Used for scalar multiplication mode 5.
VEC_MULT_RAM[1024] DSS_HWA_MULT_RAM	48	N	Vector multiplication RAM : Stores the complex vector multiplication coefficients used in modes 6 and 7.

Table 10-7. Pre-Processing Block Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
TWIDINCR	14	Y	Frequency shifter configuration: When the complex multiplication sub-block is programmed in one of the frequency shifter modes (CMULT_MODE = 0001b or 0010b), this register is used to indicate the amount of frequency shift. When the complex multiplication sub-block is programmed in FFT stitching mode (CMULT_MODE = 0011b), the last two bits of this register specify whether it is 4K or 8K FFT stitching. Specifically, if the last two bits are 01b, then it is 4K FFT stitching and if the last two bits are 10b, then it is 8K FFT stitching. Values of 00b and 11b are reserved. Also, the 12 MSB bits of this register must be kept zero in the FFT stitching mode. In all other modes of the complex multiplication sub-block, this 14-bit register must be kept as 0. When the complex multiplication sub-block is programmed with CMULT_MODE = 0110b, 0111b then the 12 MSBs of this register are used as an address offset for the Vector Multiplication Coefficients RAM (the 2 LSBs must be kept 0).
BPM_EN	1	Y	Enable/Disable BPM removal: This registerbit specifies whether the BPM removal needs to be enabled or not. If this register is set, then BPM removal is enabled prior to feeding samples from the input formatter into the core computational unit.
BPMPATTERNLSB and BPMPATTERNMSB	64	N	BPM pattern: Specifies the BPM pattern to be used to multiply the input samples if BPM removal is enabled.
BPM_RATE	10	N	BPM rate: Specifies the numberof input samples corresponding to each BPM bit. Minimum valid value for this register is 1.
BPMPHASE	4	Y	BPM startingphase: Specifies the startingphase of the BPM pattern periodicity. For more information, see the detailed description.
STG1LUTSELWR	1	N	Select Window RAM or Internal RAM: The Internal RAM for Vector Multiplication mode is mapped to the same address space as the Window RAM. Hence, this register bit is required to specify which of these two needs to be selected, when loading the co-efficients via DMA or M4. 0-Window RAM is selected 1 - Internal RAM for Vector Multiplication mode is selected. Keep this register bit as 0 always, except during the period when Internal RAM needs to be loaded.

10.2.2 CFAR Engine

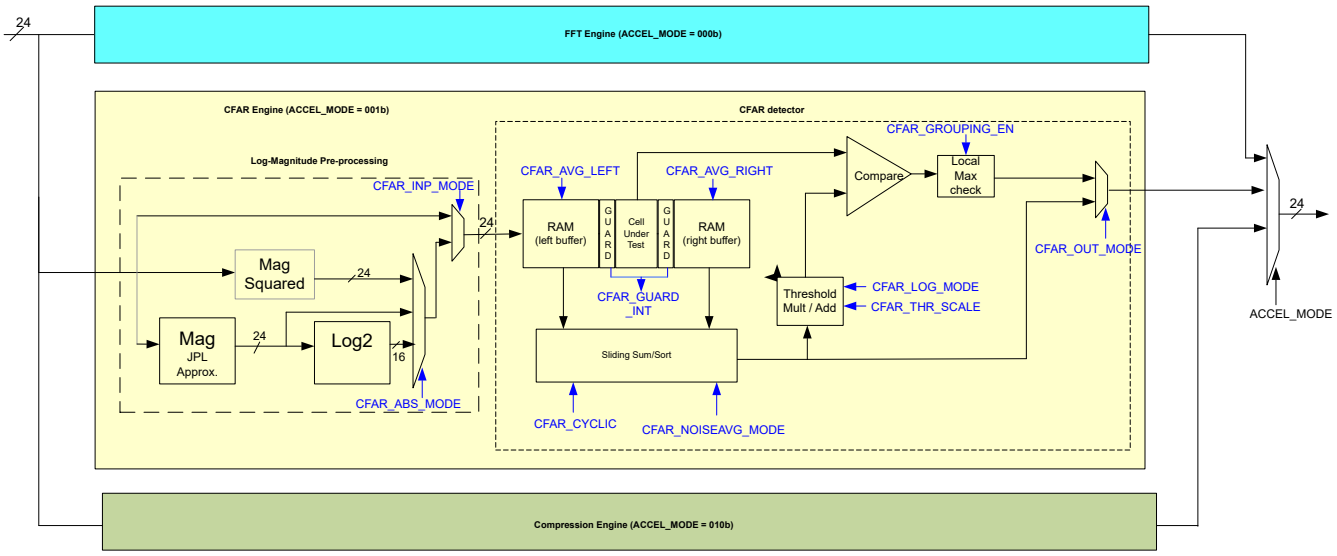


Figure 10-24. CFAR Engine

The CFAR engine (Figure 10-24) is a module that enables detection of objects, by identifying peaks in the FFT output. Although there are several detection algorithms, the accelerator supports CFAR-CA and CFAR-OS algorithms. CFAR-CA stands for constant false alarm rate – Cell Averaging. CFAR-OS stands for constant false alarm rate – Ordered Statistic.

As shown in Figure 10-24, the CFAR engine path is selected by setting the accelerator mode ACCEL_MODE = 001b. In this mode, the FFT path is not usable simultaneously and the input 24-bit samples from the input formatter block will be routed into the CFAR engine. The CFAR engine has capability to perform CFAR- detection processing (both linear and logarithmic CFAR modes are available) and generate a peak list.

In CFAR, the processing steps involve computing a threshold for each sample under test (cell under test) and deciding whether a peak is detected or not based on whether the cell under test crosses that threshold. Additionally, peak grouping may be done, where a peak is declared only if the cell under test is greater than or equal to its most immediate neighboring cells to its left and right. One thing to note here is that for peak grouping, the left and right neighboring cells themselves are not required to be CFAR qualified.

In CFAR-CA case, for each cell under test, the computation of threshold is done by averaging the magnitude (or magnitude- squared or log-magnitude) of a specified number of noise samples to the left and right of the cell under test to determine a ‘surrounding noise level’ and then applying a scale factor (or addition factor in case log-magnitude is used) on that surrounding noise average to determine the threshold. Thus, the CFAR-CA detector takes one cell at a time, computes the threshold and decides whether a valid peak is present at that cell. In the case of CFAR-OS, for each cell under test, the computation of threshold is done by sorting the magnitude (or magnitude-squared or log-magnitude) of a specified number of noise samples to the left and right of the cell under test and selecting a specific “K-th” lowest value from the sorted list as representative of the surrounding noise level, and then applying a scale factor on that value to determine the threshold.

10.2.2.1 CFAR Engine operation

The CFAR engine receives 24-bit input samples from the Input Formatter block. Typically, these are unsigned real samples, representing the magnitude or magnitude-squared or log-magnitude of the FFT output. However, the input to CFAR engine can instead be complex samples, in which case, either magnitude or magnitude-squared or log-magnitude of the complex samples can be computed inside the CFAR engine itself. This is done by the log-magnitude pre-processing sub-block inside the CFAR engine (Figure 10-25). The real unsigned result from this pre-processing operation is sent to CFAR detection processing. The registers CFAR_INP_MODE and CFAR_ABS_MODE are used to configure real vs. complex input, as well as the nature of pre-processing required (refer register description table). The log-magnitude computation uses the same JPL approximation for magnitude calculation and the same look-up table (LUT) approximation for log2 computation as described in Part

1 of the user guide for FFT engine post-processing. Note that for the case of real input (i.e., CFAR_INP_MODE = 1), the input samples must be unsigned. In this case, CFAR_ABS_MODE register has no effect.f

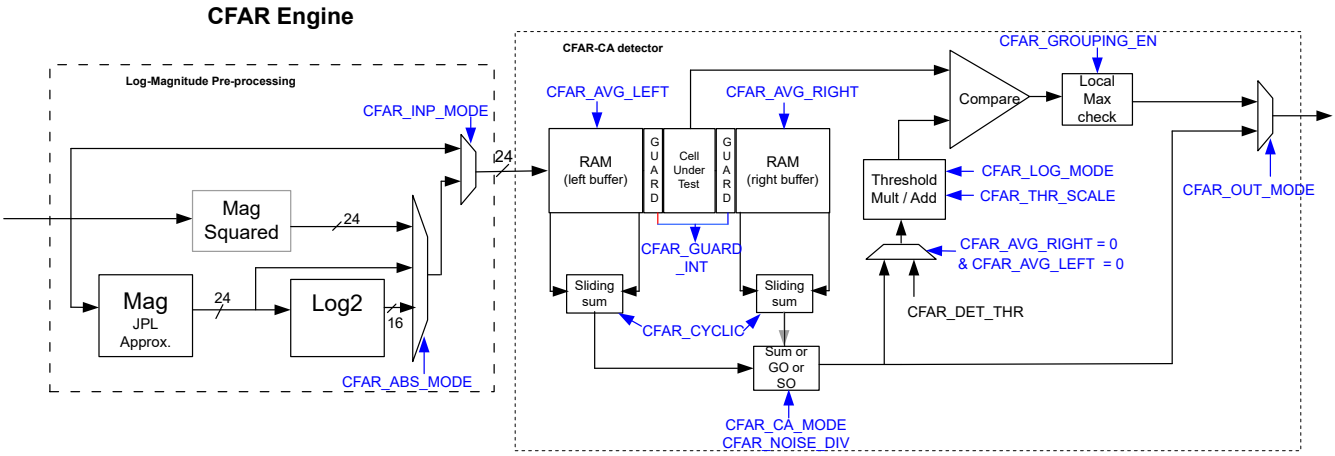


Figure 10-25. CFAR-CA block diagram

As described earlier, the CFAR- detection processing involves finding a “surrounding noise level” for each cell under test and then determining a threshold that is a function of the surrounding noise level. The cell under test is compared against this threshold to decide whether a peak is present or not in that cell. To calculate the threshold, the surrounding noise level is multiplied with (or added to) a threshold scaling factor specified in CFAR_THRESH register. There are two modes in which the CFAR detector can be used – in non-logarithmic mode (a.k.a linear CFAR), the threshold scale factor is multiplied, and in logarithmic mode (a.k.a logarithmic CFAR), the threshold scale factor is added. This is decided based on CFAR_LOG_MODE register.

Note: The linear and logarithmic modes are available for CFAR-CA and its variants. Their detection cores are built with 24-bit datapath width. Only the logarithmic mode is available for CFAR-OS. The CFAR-OS detection core is built with 16-bit datapath width, which is sufficient in logarithmic mode. The final detection threshold that is so obtained is used to compare against the cell under test to determine whether a peak is detected in that cell.

Table below summarizes the register settings for the different CFAR modes of operation. The surrounding noise level computation has multiple options – cell averaging (CFAR-CA), cell averaging with greater-of selection (CFAR-CAGO), cell averaging with smaller-of selection (CFAR- CASO) and ordered statistic (CFAR-OS). The register CFAR_CA_MODE is used to select one among CFAR-CA, CFAR-CAGO, CFAR- CASO and CFAR-OS modes. In CFAR-CA, the noise samples on the left side and right side of the cell under test (after ignoring some guard cells on either side) are simply averaged to determine the surrounding noise level. In CFAR-CAGO, the noise samples on the left side and right side are averaged independently and the greater of the two is used to determine the threshold. In CFAR-CASO, the lesser of the two is used. In CFAR-OS, the noise samples on the left side and right side of the cell under test (after ignoring some guard cells on either side) are sorted and the “K-th” lowest value from the sorted list is selected as the surrounding noise level. The selection of “K-th value” from the sorted list is based on the register CFAR_OS_KVALUE (see table of registers for more details on this register).

Table 10-8. CFAR modes and Register settings

DesiredCFAR Mode	InputRealor Complex	DesiredPre-Processing	RegisterValues to Use		
			CFAR_INP_MODE	CFAR_ABS_MODE	CFAR_LOG_MODE
LinearCFAR	Real	N/A	1	00	0
	Complex	Magnitude	0	10	0
		Mag-squared	0	00	0
		Log2-Mag	0	11	0
LogCFAR	Real	N/A	1	00	1
	Complex	Log2-Mag	0	11	1

DesiredCFAR Algorithm	CFAR_CA_MODERegister Setting
CFAR-CA	00
CFAR-CAGO	01
CFAR-CASO	10
CFAR-OS	11

The number of samples on the left side and right side used for computing the noise average is configured using CFAR_AVG_LEFT and CFAR_AVG_RIGHT registers and the number of guard cells is configured using CFAR_GUARD_INT register (Figure 10-25). The number of samples used for left side noise averaging is given by 2*CFAR_AVG_LEFT. The number of samples used for right side noise averaging is given by 2*CFAR_AVG_RIGHT. The number of guard cells that are ignored on each side of the cell under test is given by CFAR_GUARD_INT. For example as shown in Figure 10-31, if CFAR_AVG_LEFT = CFAR_AVG_RIGHT = 16, and CFAR_GUARD_INT = 3, then it means that the most immediate three samples each to the left and right of the cell under test are skipped and then, 32 samples on the left and 32 samples on the right side are used for noise averaging. Note that even though the term noise averaging is used here, the actual implementation simply adds the noise samples first and the “averaging” is done as a divide by a power-of- 2 as specified in a separate register, CFAR_NOISE_DIV. These registers are described in Table of register descriptions.

Note: The CFAR engine also supports a special “constant threshold mode” of CFAR detection. In this special mode, the detection threshold value to compare with each cell-under-test is based on a user configurable constant – CFAR_DET_THR. This detection threshold value is independent of “surrounding noise level”, and the detection comparison depends only CFAR_DET_THR, CFAR_THR_SCALE, and CFAR_LOG_MODE. This mode of operation can be achieved by setting the engine in CFAR-CA mode and additionally setting CFAR_AVG_LEFT = CFAR_AVG_RIGHT = 0. In this constant threshold mode, CFAR_THRESH scale factor is multiplied with CFAR_DET_THR in the linear mode, and in the logarithmic mode the threshold scale factor is added.

In case of CFAR-CA, the valid values for CFAR_AVG_LEFT and CFAR_AVG_RIGHT is any number between 0 and 63 (except 1), which means that the number of samples each on the left side and right side used for noise averaging can be one of 0, 4, 6, 8, 10, 12, 14, … 124, 126. The values of CFAR_AVG_LEFT and CFAR_AVG_RIGHT can be different in cyclic mode of CFAR and need to be equal in non-cyclic mode (both are described in a later section).

However, in the case of CFAR-OS, the valid values for CFAR_AVG_LEFT and CFAR_AVG_RIGHT are highly restricted. They need to be equal (i.e., same window size on left and right sides) and further, the only values supported for these registers in CFAR-OS mode are: 0, 4, 6, 8, 12, 16, 24 and 32 (which corresponds to number of samples being 0, 8, 12, 16, 24, 32, 48 and 64). Note that the register CFAR_NOISE_DIV, which is used in CFAR-CA for noise “averaging”, is not applicable in case of CFAR-OS.

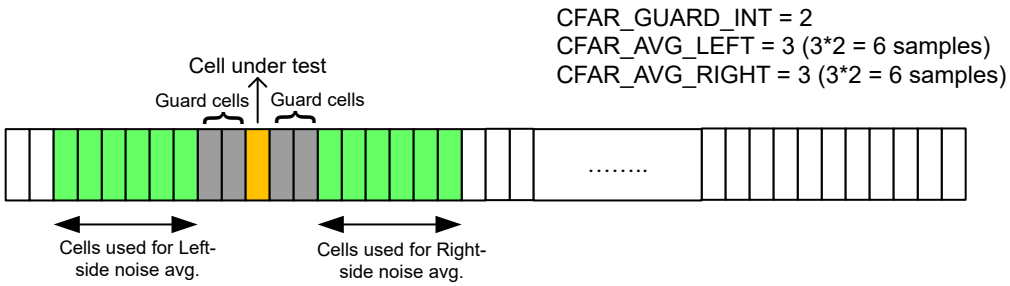


Figure 10-26. CFAR-CA: Cells used for Surrounding Noise Average

As mentioned earlier, the CFAR_THRESH register specifies the threshold scaling factor. This is an 18-bit register whose value is used to either multiply or add to the ‘surrounding noise level’ to determine the threshold used for detection of the present cell under test. If logarithmic mode is disabled (in magnitude or magnitude-squared mode), then the register value is multiplied with the surrounding noise level to determine the threshold, else it is added to the surrounding noise level. In the former case, this 18-bit register is interpreted as a 14.4 value

and supports a range of values from 1/16 to 2^14-1. In the latter case (logarithmic mode), the 18-bit register is interpreted as a 7.11 value.

The CFAR engine supports a few output formats that are described next

10.2.2.2 CFAR Engine Output Formats

As part of CFAR detection, the cells that exceed the threshold are noted and this 'Detected Peaks list' is sent to the destination memory. Since the output format of the core computational unit is 24-bits I and 24-bits Q, the detected peaks list is formatted into 'I' and 'Q' channels as shown in Figure 10-27. The 24-bit I channel contains the index at which the peak is detected, with the MSB 12 bits containing the iteration number (corresponding to BCNT counter value) and the LSB 12 bits containing the sample index number (corresponding to SRCACNT counter value). The 24-bit Q channel contains the surrounding noise level value or the cell under test value of that detected peak. This is chosen based on CFAR_OUT_MODE register setting. Instead of 'Detected Peaks list', it is also possible for the CFAR engine to send out the raw 'surrounding noise level' value for each cell. This is called 'Raw output mode'.

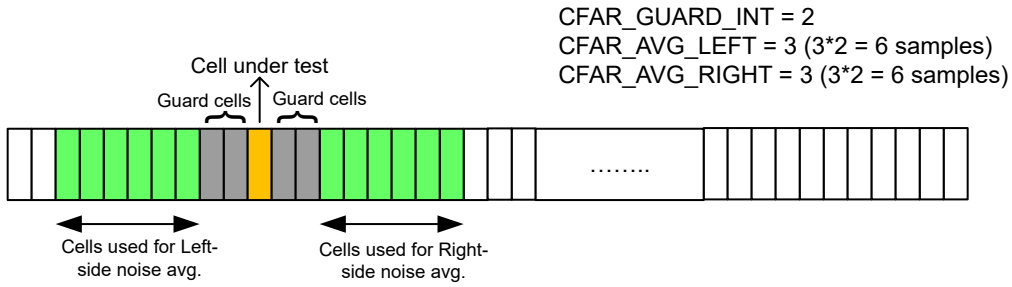


Figure 10-27. CFAR Engine Output Format

In detected peaks list mode, only the detected peaks are output to the destination memory. In this case, the read-only register CFARPEAKCNT indicates how many peaks have been totally detected, so that the main processor can read that many locations from the destination memory. In this mode, the number of peaks stored in the destination memory is limited to a maximum of 4095, or DSTACNT, whichever is smaller. If more peaks are detected beyond this number, they wrap around and circularly overwrite the same locations in the destination memory. Also, in this mode, the register DSTBINDX is not applicable and is ignored.

While detecting peaks, if 'peak grouping' is required, then it can be enabled using CFAR_GROUPING_EN register. In this case, a peak is declared as detected only if it the cell under test exceeds the threshold, as well as, if the cell under test exceeds the two neighboring cells to its immediate left and right (the peak is a local maximum).

Table 10-9. CFAR Output modes and Register Settings

CFAROutput Mode	IChannel Output	QChannel Output	Register Settings (CFAR_ADV_OUT_MODE, CFAR_OUT_MODE)
Rawoutput mode (all cells are output)	Surrounding noiselevel	Cellunder test value	(0,00)
	Surrounding noiselevel	Binary detectionresult flag (0 or 1)	(0,01)
Detectedpeaks list mode (only detected peaks are output)	Peak index	Surrounding noiselevel value	(0,10)
	Peak index	Cellunder test value	(0,11)

10.2.2.3 CFAR Engine Cyclic vs. Non-Cyclic

The register CFAR_CYCLIC specifies whether the CFAR detector needs to work in cyclic mode or in non-cyclic mode. In general, the programmed number of samples for noise level computation (specified by CFAR_AVG_RIGHT and CFAR_AVG_LEFT) are available fully only for the cells under test which are in the middle of the input array (Figure 10-28). For first several cells under test, the available number of samples to the left is lesser than the programmed number. Similarly, for the last several cells under test, the available number of samples to the right is lesser than programmed.

In cyclic mode (Figure 10-29), this is handled by wrapping around the edges in a circular manner. For a cell under test near the left edge, some samples from the right edge (circular wrap around the edge) are fetched to collect the programmed CFAR_AVG_LEFT number of left side samples for noise level computation. Similarly, for a cell under test near the right edge, an appropriate number of samples from the left edge are used (again, circular wrap around the edge).

This cyclic CFAR implementation is accomplished through a combination of a few register settings within the CFAR engine, as well as in the input and output formatter blocks. Specifically, the input formatter is configured to send additional samples (repeat samples) in a circular manner wrapping around the left and right edges. This is achieved by using the A-dimension circular shift (SRCA_CIRCSHIFT) and wrap-around (SRCA_CIRCSHIFTWRAP) registers in the input formatter, such that the required number of extra samples at both edges are streamed into the CFAR engine. The cyclic CFAR mode only works when the number of cells under test is a power of 2.

For example (Figure 10-30), if the number of cells under test is 256, the average number of left and right noise samples is 32 each and the number of guard cells is 3 on either side. Then, the registers need to be programmed as shown in Table 5.

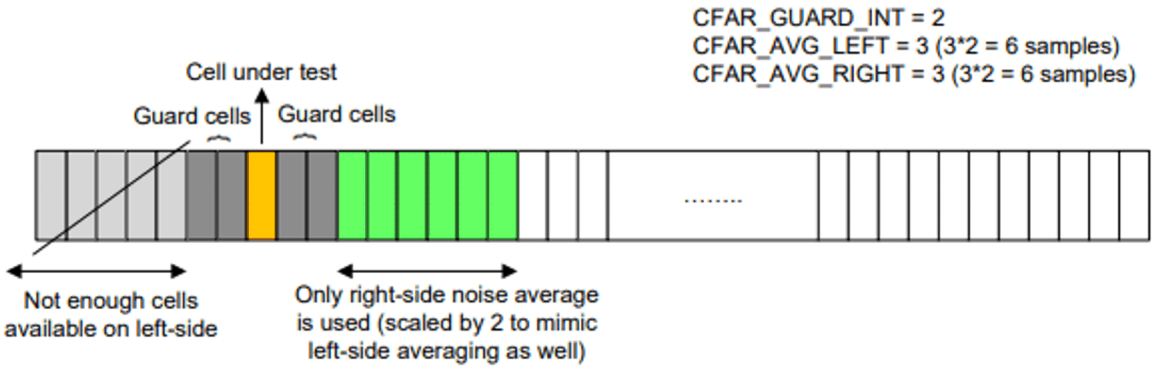


Figure 10-28. Handling of samples near edges in non-cyclic mode

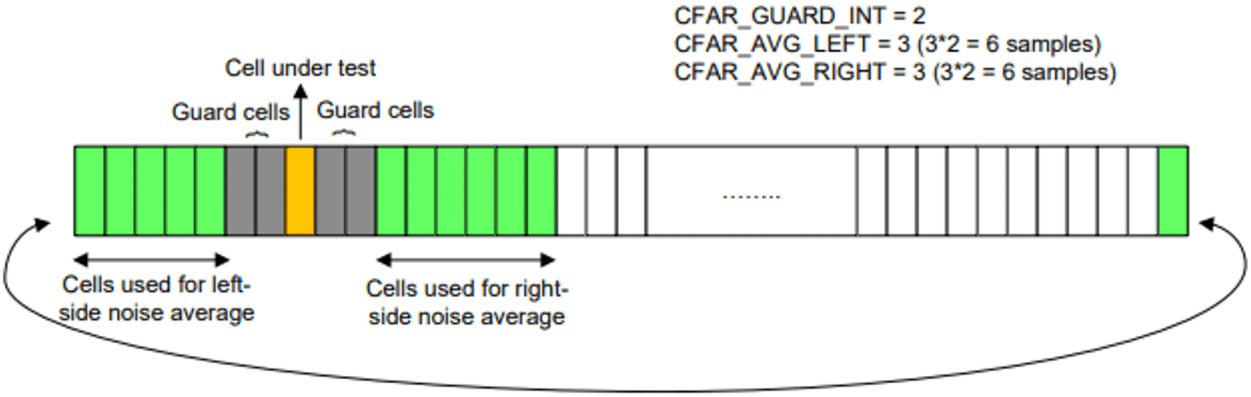


Figure 10-29. Handling of samples near edges in cyclic mode

Table 10-10. Configuration example for CFAR Cyclic mode

Module	RegisterSetting	Comments
CFAREngine	CFAR_GUARD_INT= 3	3guard cells on either side
	CFAR_AVG_LEFT= 16 CFAR_AVG_RIGHT= 16	32samples on left side and 32 samples on right side for noise averaging
Input Formatter	SRCACNT =325	255+ (32+3) + (32+3), where 255 is the usually configured value of SRCACNT for a 256 sample vector, plus 32+3 additional samples for circular repeat at either end
	SRCA_CIRCSHIFT= 221	256– (32+3), which is the starting offset for the circular shift, so that samples are streamed into CFAR engine start from this point
	SRCA_CIRCSHIFTWRAP= 8 SRC_CIRCSHIFTWRAP3X = 0	The circularwrap-around happens when SRCACNT counter value reaches 2^SRCA_CIRCSHIFTWRAP = 256
Output Formatter	REG_DST_SKIP_INIT= 0	Noneed to skip any samples at Output Formatter even though extra samples are fed into CFAR engine, because CFAR engine automatically strips out the extra samples
	DSTACNT= 255	256outputs corresponding to 256 cells

However, the handling of edge samples in non-cyclic mode of CFAR is different. It is explained below – first for CFAR-CA and then for CFAR-OS versions.

In non-cyclic mode of CFAR-CA, if the number of available samples on the left for any cell under test is lesser than CFAR_AVG_LEFT, then the noise average is computed solely from the right side. This is done by calculating the noise sum as twice the right side noise sum. Similarly, if the number of available samples on the right is lesser than CFAR_AVG_RIGHT, then the noise average is computed solely from the left side. This is done by calculating the noise sum as twice the left side noise sum. It is required that the CFAR_AVG_LEFT and CFAR_AVG_RIGHT be programmed equally in non-cyclic mode – otherwise, the noise computation for the edge samples is not ideal.

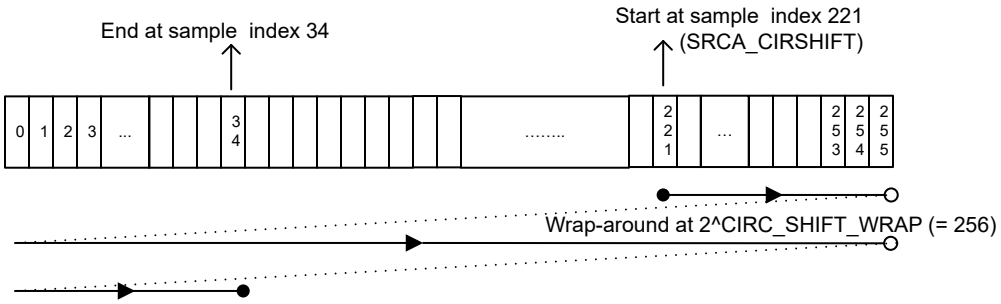


Figure 10-30. Input Formatter Streaming for Cyclic CFAR example

In non-cyclic mode of CFAR-OS, the edge samples are handled as follows. For the cells under test that are near the edges, the number of available surrounding samples for sorting is lesser than programmed (CFAR_AVG_LEFT or CFAR_AVG_RIGHT). These available samples are first sorted and the Kth lowest value is selected as the noise level. It should be noted that this Kth lowest sample in the available samples may result in a sub-optimal noise level for edge samples than non-edge samples because the number of available samples is lesser for edge samples than for non-edge samples. A minor variant for better handling this edge sample case can be enabled by setting the register CFAR_OS_NONCYC_VARIANT_EN to 1. In that case, available samples are first sorted. But instead of using the programmed K value directly for noise sample selection from the sorted array, a proportionally scaled down value is used based on the number of available surrounding samples for each cell under test. This is illustrated in Table below, where, L represents the programmed CFAR_AVG_LEFT (same as CFAR_AVG_RIGHT) and K represents the programmed CFAR_OS_KVAL. It is required that the CFAR_AVG_LEFT and CFAR_AVG_RIGHT be programmed equally in non-cyclic mode.

Table 10-11. Internal K value used in CFAR-OS non-cyclic mode

No. of available samples on one side (excluding guard)	No. of available samples on the other side (excluding guard)	Internal K value used for noise sample selection (CFAR_OS_NON_CYC_ VARIANT_EN = 0)	Internal K value used for noise sample selection (CFAR_OS_NON_CYC_ VARIANT_EN = 1)
L	0 to floor(L/4)-1	K	floor (4K/8)
L	floor(L/4) to floor(2L/4)-1	K	floor (5K/8)
L	floor(2L/4) to floor(3L/4)-1	K	floor (6K/8)
L	floor(3L/4) to floor(4L/4)-1	K	floor (7K/8)
L	L	K	K

In general, it is expected that CFAR Engine will be used for arrays much larger than the configured left and right window and guard lengths. Specifically, the ACNT should exceed the sum of configured left and right window and guard lengths.

10.2.2.4 CFAR Engine Register Descriptions

CFAR Engine Registers lists all the registers of the CFAR engine block.

Table 10-12. CFAR Engine Registers

Register.field	Width	Parameter-Set? (Y/N)	Description
CFAR_AVG_LEFT	6	Y	Numberof left-side samples for noise level computation: This registeris used to specify the number of samples used for noise level computation to the left of the cell under test. The number of samples used for noise level computation is equal to the value of this register multiplied by 2. For example, if this register value is 15, then the number of left- side samples used for averaging is 30. The maximum number that is possible is 126. A value of zero in this register means that the noise samples on the left side are not used for noise level computation. The valid values for this register are different for CFAR-CA and CFAR-OS modes: In CFAR-CA (and its variants CFAR-CAGO and CFAR-CASO), valid values for this register are 0, 2, 3, 4, …63 (Note that avalue of 1 is not supported). This corresponds to number of samples equal to 0, 4, 6, 8, 10, 12, 14, … 124, or 126. In CFAR-OS mode, valid values for this register are restricted to 0, 4, 6, 8, 12, 16, 24, 32 only (which corresponds to number of samples equal to 0, 8, 12, 16, 24, 32, 48 or 64).
CFAR_AVG_RIGHT	6	Y	Numberof right-side samples for noise level computation: This registeris very similar to the above, except that this register specifies the averaging to the right of the cell under test. In most cases, it is expected that CFAR_AVG_RIGHT has the same value as CFAR_AVG_LEFT. In non-cyclic modes of CFAR, CFAR_AVG_RIGHT must be programmed equal to CFAR_AVG_LEFT.

Table 10-12. CFAR Engine Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
CFAR_GUARD_INT	3	Y	Numberof guard cells: This registerspecifies the number of guard cells to ignore on either side of the cell under test. If this register value is 3, then three guard cells on the left side and three guard cells on the right side are ignored. Only the noise samples beyond this guard region are used for calculating the surrounding noise level.
CFAR_OS_KVALUE	6	Y	K-th value for ordered statistic: This register is useful only in CFAR-OS mode, where it indicates the parameter K. From the sorted list of left and right noise samples, the K'th lowest value is used as the noise sample. This is a zero-based count – for instance, if this register value is 27, then the 28^{th} lowest element in the sorted array is selected. Note that since CFAR-OS supports a maximum of 32 samples each on left and right side, the maximum size of the vector to sort is 64, and hence the maximum valid value of CFAR_OS_KVALUE register is 63.
CFAR_OS_NON_CYC_VARIANT_EN	1	Y	Enable scaling of K value for edge samples in non-cyclic CFAR-OS: This is useful only in CFAR-OS in non-cyclic mode. Setting this to 1 enables a variant where the K value used for noise sample selection for edge samples is scaled down proportional to the number of available neighboring samples at edges.
CFAR_THRESH	18	N	Threshold scalefactor: This registeris used to specify the threshold scale factor. This value is used to either multiply or add to the 'surrounding noise level' to determine the threshold used for detection of the present cell under test. If logarithmic CFAR mode is disabled (in magnitude or magnitude-squared mode), then the register value is multiplied with the surrounding noise level to determine the threshold, else it is added to the surrounding noise level. In the former case, this 18-bit register is interpreted as a 14.4 value. In the latter case (logarithmic mode), the 18-bit register is interpreted as a 7.11 value.
CFAR_DET_THRESH	24	N	Constant detection threshold value in constant threshold mode: This register is applicable only in constant threshold mode of CFAR (i.e. only in CFAR-CA mode and only if CFAR_AVG_LEFT = CFAR_AVG_RIGHT = 0). In this special mode, this register specifies the detection threshold value used to compare with cell under test. The detection threshold value is held constant, and scaled by CFAR_THRESH linearly or logarithmically

Table 10-12. CFAR Engine Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
CFAR_LOG_MODE	1	Y	CFARlinear or logarithmic mode: This registeris one of the registers used to specify whether the CFAR detector operates in linear or logarithmic mode. If this register bit is set, then the CFAR detector operates in logarithmic mode, which means that the threshold scale factor is added to (instead of multiplied with) the surrounding noise level value to determine the threshold. Note that this mode is meaningful only when the input samples to the CFAR detector are log- magnitude samples (see CFAR_INP_MODE as well). If this register bit is 0, then the logarithmic mode is disabled, in which case, the threshold scale factor is multiplied with (instead of added to) the surrounding noise level to determine the threshold. This mode is meaningful when magnitude or magnitude-squared samples are fed to the CFAR detector.
CFAR_INP_MODE	1	Y	CFARengine input mode: This registerbit specifies whether the inputs to the CFAR engine are complex samples or real values (the real values are already magnitude, magnitude-squared or log-magnitude numbers that can be directly sent to CFAR detection process). If this register bit is 1, then the input samples are real values and are directly sent to CFAR detection. If this register bit is 0, then the inputs are complex samples and hence either magnitude or magnitude-squared or log-magnitude computation is required prior to CFAR detection. Which of the three, viz., magnitude or magnitude-squared or log-magnitude is done, is selected by CFAR_ABS_MODE register described below.
CFAR_ABS_MODE	2	Y	CFARmagnitude, mag-squared or log-mag mode: This registeris used to specify which of the three computations, namely Magnitude, Mag- squared or Log-Magnitude, is enabled inside the CFAR engine prior to CFAR detection. This register is only relevant when CFAR_INP_MODE is 0 (complex samples are fed to CFAR engine). 00b– Magnitude-squared 01b – Not valid 10b– Magnitude (using JPL approximation) 11b– Log2-Magnitude (using LUT approximation)

Table 10-12. CFAR Engine Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
CFAR_OUT_MODE	2	Y	CFARengine output mode: The MSBbit of this register selects whether the CFAR Engine outputs all the noise average values for all the cells (‘Raw output’ mode), or whether the CFAR Engine outputs only the detected peaks (‘Detected Peaks List’ mode). The LSB bit specifies the content of the 24-bit ‘I’ and ‘Q’ channel outputs logged in destination memory. If CFAR_ADV_OUT_MODE is set to 1 (special mode called “Dominant peaks”), this register should be set to 1. Refermain description section for details.
CFAR_GROUPING_ EN	1	Y	CFARpeak grouping enable: This registerbit specifies whether peak grouping should be enabled. When this register bit is 0, peak grouping is disabled, which means that a peak is declared as detected as long as the cell under test exceeds the threshold. On the other hand, if this register bit is 1, then a peak is declared as detected only if it the cell under test exceeds the threshold, as well as, if the cell under test exceeds the two neighboring cells to its immediate left and right (local maximum).
CFAR_NOISE_DIV	4	Y	CFARnoise average division factor: This parameter is applicable only in CFAR-CA modes and it is not applicable in CFAR-OS mode. This registerspecifies the division factor with which the noise sum calculated from the left and right noise windows are divided, in order to get the final surrounding noise average value. The division factor is equal to 2^ CFAR_NOISE_DIV. Therefore, only powers-of-2 division are possible, even though the number of samples specified in CFAR_AVG_LEFT and CFAR_AVG_RIGHT are not restricted to powers of 2. The surrounding noise average value obtained after the division is multiplied or added with CFAR_THRESH to determine the final threshold used to compare the cell under test for detection. The maximum allowed value for this register is 8, which gives a division factor of 256.
CFAR_CA_MODE	2	Y	CFARnoise averaging mode: This registerconfigures the noise averaging mode in the CFAR detector from one of these options – CFAR-CA, CFAR-CAGO, CFAR-CASO, CFAR-OS. 00b– CFAR-CA 01b– CFAR-CAGO 10b– CFAR-CASO 11b– CFAR-OS

Table 10-12. CFAR Engine Registers (continued)

Register.field	Width	Parameter-Set? (Y/N)	Description
CFAR_CYCLIC	1	Y	CFARcyclic vs. non-cyclic mode: This registerbit specifies whether the CFAR detector needs to work in cyclic mode or in non-cyclic mode. When this register bit is 0, the CFAR detector works in non-cyclic mode and when it is 1, it works in cyclic mode. Refer main description section for details on how to configure and use cyclic mode.
CFAR_PEAKCNT	12	N	CFARdetected peak count: This isa read-only register that contains the number of detected peaks that are logged in the destination memory, when CFAR Engine is configured in 'Detected Peaks List' mode. In the Detected Peaks List mode, since only the detected peaks are logged in the destination memory, this read-only register provides the number of detected peaks that are logged to the main processor, so that the main processor can determine how many entries to read from the destination memory.

10.2.3 FFT Engine – Statistics

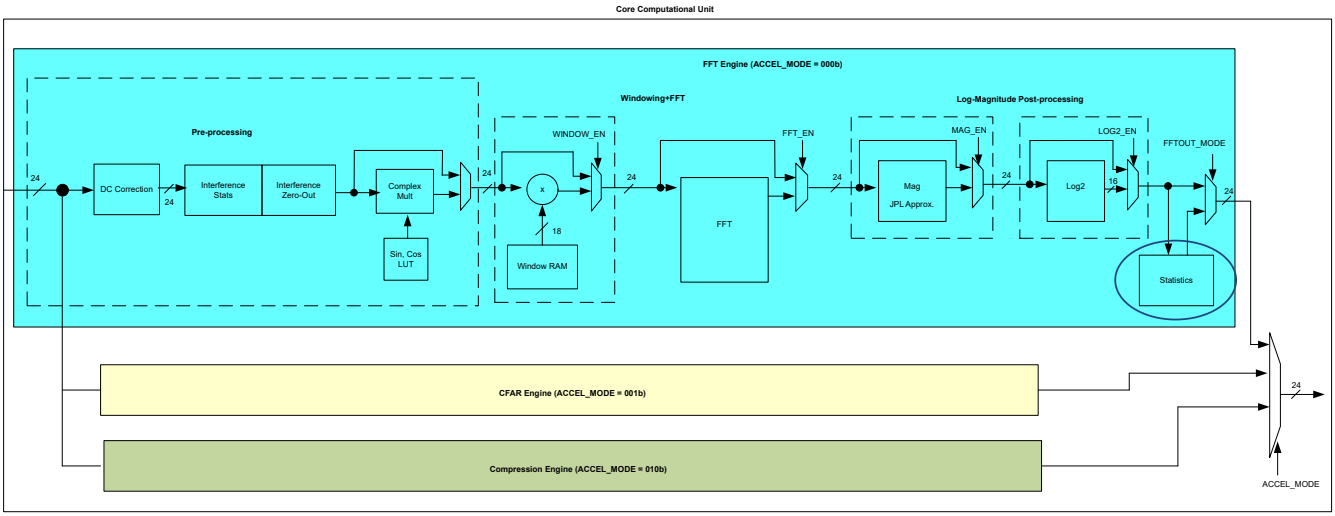


Figure 10-31. Statistics Block

The core computational unit has a statistics computation block at the end of the FFT Engine path as shown in Figure 10-31. It can be used to compute a few simple statistics of the samples output by the core computational unit. It supports computing statistics on vector inputs - it can compute sum and maximum of samples.

10.2.3.1 Statistics Block – Operation

The 24-bit I and 24-bit Q output of the core computational unit goes to a statistics computation block. The purpose of this block is to find the maximum and sum (average) of the output samples

The sum and max statistics are computed on a 'per-iteration' basis (the sum and max values are logged at the end of each iteration) and the computation is reset for the next iteration. The sum and max values are logged in register-sets (see MAXn_VALUE, MAXn_INDEX, ISUMn, QSUMn register-sets), which can be read by the main processor. However, only four such registers are provided for each statistic and therefore, the sum and max values can be logged in these registers only for up to a maximum of four iterations.

The max statistics register-set comprises four read-only registers of 24 bits each, named MAXn_VALUE, for recording max values, and four read-only registers each 12 bits unsigned, named MAXn_INDEX, for recording the max indices. The sum statistics register-set contains four registers of 36 bits each, named ISUMn, for I-sum statistics, and 4 registers of 36 bits each, named QSUMn, for Q-sum statistics.

For larger number (>4) of iterations, either the sum or the max value can be sent to the destination memory for each iteration, which allows the statistic to be available even for cases with more than four iterations. The logging of the statistic into the destination memory is enabled using FFT_OUTPUT_MODE register described below.

The MSB bit of the FFT_OUTPUT_MODE (Table below) register selects whether the default (main) output of the core computational unit goes to the destination memory, or the statistics block output. If the MSB of this 2-bit register is 0, then it selects the default mode of operation, where the main output (FFT or Log-Mag result) is sent to the destination memory. If the MSB is 1, then it selects the statistics output mode, where either the sum or max statistic is sent to the destination memory (one value per iteration). Whether the sum or max is sent to memory is dependent on the LSB bit. If the LSB bit is 0, then the statistic value that is sent is the max value (useful in conjunction with Log-Mag enabled to find the biggest peak and peak index per iteration). Here, the I output is the maximum value itself and the Q output is the index (location) of the maximum value. If the LSB bit is 1, then the statistic value that is sent is the sum value (useful for DFT mode, as well as for mean squared or mean of absolute values computation).

Table 10-13. Statistics Output Modes

FFT_OUTPUT_MODERegister	IChannel Output	QChannel Output
00b– Default output mode	Mainoutput of core computational unit	
10b– Max statistics output (One output per iteration)	MaxValue	MaxIndex
11b– Sum statistics output (One output per iteration)	Sum ofI values	Sum ofQ values

The max statistic records the maximum value (and its index) of the magnitude or log-magnitude samples corresponding to every iteration. The sum statistic records the sum of the magnitude or log-magnitude or the complex output samples corresponding to each iteration. If the main output of the core computational unit is the complex FFT output (ABS EN=0 and LOG2EN=0), then the sum statistics is the complex sum.

The complex sum statistics mode is useful when used in conjunction with the complex multiplier block in DFT mode or vector multiplication mode. For example, the sum statistic computed here, together with the DFT mode of the complex multiplier block, enable DFT computation for the desired number of bins (iterations). When the desired number of bins is more than 4, the sum statistic can be sent to destination memory (instead of the main data output that is normally sent to the destination memory).

Note that when the sum statistics is logged into the destination memory, it goes through the Output Formatter block as only 24-bits each for I and Q (same bit-width as the primary FFT outputs). Hence, the computed sum statistics value of 36-bits width, needs to be scaled down by right-shifting the appropriate number of LSBs (using FFTSUMDIV register) before sending to output formatter. Thus, when logging the statistics in destination memory, the sum statistics is to be used as an “average” value, rather than a “sum” value itself.

The FFTSUMDIV register specifies the number of bits to right-shift the sum statistic before it is written to destination memory. The internal sum statistic register is 36-bits wide (allowing 12 bits of MSB growth of the 24-bit data path), but this statistics value needs to be scaled down to 24 bits to match the data path width going to the Output Formatter. This register specifies how many LSBs to drop to convert the sum statistics to 24-bit value. Note that only signed saturation is implemented (irrespective of whether magnitude values are being summed or complex FFT output values are being summed). Therefore, it is recommended that this register is configured to drop an appropriate number of LSBs such that incorrect saturation in case of magnitude sum is avoided.

Note that in statistics output mode, the registers DSTACNT, DSTAINDX, DSTBINDX, DST16b32b and DSTREAL are not meant to be used, since it is known that there is only one value to be written to destination memory for every iteration in a specific format. It is recommended that in this mode, DSTACNT be programmed to a

value of 0, DSTAINDX and DSTBINDX are both programmed to a value of 8 bytes, DST16b32b is set to 1 and DSTREAL is reset to 0. The statistics is then always logged in the destination memory as consecutive 32-bit I and Q samples, irrespective of whether sum statistic or max statistic is being logged.

10.2.3.2 Statistics Block – Register Descriptions

Statistics Block Registers below listsall the registers of the statistics block.

Table 10-14. Statistics Block Registers

Register.field	Width	Parameter-Set?(Y/N)	Description
MAX<n>_VALUE n=1..4	24	N	Maxvalue: These registerscontain the max value on a per-iteration basis. These registers are meaningful only when Magnitude or Log-Magnitude is enabled. Only the max values for up to four iterations are recorded in these registers. For larger number of iterations, use statistics output mode (FFT_OUTPUT_MODE below).
MAX<n>_INDEX n=1..4	12	N	Maxindex: These registerscontain the max index on a per-iteration basis, corresponding to each max value in the MAXn_VALUE registers.
I_SUM<n>_LSB I_SUM<n>_MSB Q_SUM<n>_LSB Q_SUM<n>_MSB			Sum statistics: These registerscontain the sum of the I outputs and Q outputs on a per-iteration basis. Only the statistics for up to four iterations are recorded in these registers. For larger number of iterations, use statistics output mode (FFT_OUTPUT_MODE below).
FFT_OUTPUT_ MODE	2	Y	FFT Path outputmode: This registerspecifies the output mode of the FFT path. Instead of the default mode where the main output of the core computational unit is sent to the destination memory, this register can be configured such that either the max or sum statistics can be sent to the destination memory. 00b– Default mode (main output) 10b – Max statistics output mode 11b – Sum statistics output mode
FFTSUMDIV	5	N	Right-shiftingfor Sum statistic: This registerspecifies the number of bits to right-shift the sum statistic before it is written to destination memory. The internal sum statistic register is 36-bits wide (allowing 12 bits of MSB growth of the 24-bit data path), but this statistics value needs to be scaled down to 24 bits to match the data path width going to the Output Formatter. This register specifies how many LSBs to drop to convert the sum statistics to 24-bit value.

10.3 RadarHardware Accelerator - Part 3

This part of the HWA user guide is organized as follows:

- Section 1 provides an overview of the features of the compression engine.
- Section 2 describes the compression algorithms available in the engine.
- Section 3 covers information related to configuration and register descriptions.

10.3.1 Compression/Decompression Engine Overview

The Compressionand Decompression Engine (referred henceforth simply as the ‘compression engine’) consists of

- a compression engine which takes a fixed number of samples and returns a ‘block of bits’ such that the block’s size (in bits occupied) is a fraction of the size of the input samples, and
- a decompression engine which when provided with the same compressed block of bits, regenerates the original samples (with the possibility of some quantization error).

The features of the compression engine are

- The compression engine is designed to achieve arbitrary compression ratio. (‘compression ratio’ is defined as the ratio of ‘average bit-width per sample’ after compression and ‘the original bit-width’ before compression. In other words, a 33 % compression-ratio, results in the average bitwidth after compression being 1/3rd of the bitwidth before compression.)
- It has two configurable algorithms for compression and decompression.
  - Block Floating Point (BFP).
  - Exponential-Golomb Encoding (EGE).
- It implements a ‘block’ based compression scheme - i.e. it takes a fixed number of samples (called a block) and creates a ‘compressed block of bits’ of fixed size. During the Doppler processing operation, when radar data has to be accessed or written in transpose, having each block as a fixed size simplifies the EDMA programming. The EDMA can simply access a full block (across Doppler) in much the same manner as a single range gate.
- It is a part of the HWA as one of the programmable ‘paths’ in the accelerator (in addition to the FFT and CFAR paths). It can therefore use existing capabilities/resources of the HWA (input/output formatters, state machine, looping, etc).

Note

Note: Low compression ratios can result in ‘high quantization noise’. Designers should select the appropriate ‘compression ratio’ after confirming that it meets the dynamic range necessary for their application.

10.3.2 Algorithms

The following section is a brief introduction to the two algorithms used in the compression engine.

10.3.2.1 Block Floating Point (BFP)

The block floating point compression algorithm is a simple method by which a block of samples are given a common exponent (referred to as scalefactor with bits) based on the largest sample in the block. Each sample in the block is also assigned a fixed number of mantissa bits (with bits). The size of the compressed block is then simply . The compression ratio can be tweaked by varying . Smaller have better compression ratios but higher quantization noise.

As an example, consider a block of 2 samples, where each sample is a 32-bit long complex number (i.e. 16 bit I, 16 bit Q), it is possible to compress the block by giving it a common scale factor, b(of width b_w = 4 bits) and then encoding the remaining data as 7 bit mantissas (i.e. 14 bits for complex number). The total compressed block size is 32 bits (2*14+4 = 32 bits). This configuration thus enables compression scenario of 50%. Also, with 7 bits of mantissa, the dynamic-range preserved per block of samples is ≈ 7 × 6 dB (or 42 dB).

The decompression engine takes a compressed block of data and regenerates the original samples (with some quantization error). It extracts the scalefactor (b), scales the mantissa and place it in an output buffer. Put simply, the sample is reconstructed as '$mantissa \times 2^b$'.

10.3.2.1.1 The BFP format

The block starts with an optional header holding the scale factor. The rest of the block is filled with the mantissa of each sample directly in twos-complement format (since the input samples are signed numbers). The scale factor and the mantissa have known bitwidths.

There may be empty space (for padding) at the end of the compressed block, in case the compressed block size is not a multiple of a byte/word. The exact number of padding bits that are present would depend on the desired compression, based on the size of the mantissa and the size of the header and the number of samples per block.

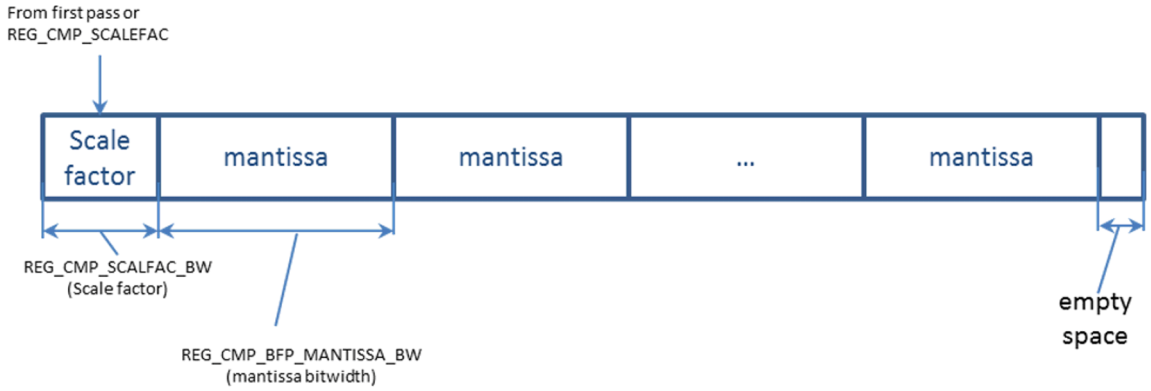


Figure 10-32. BFP – format

10.3.2.2 Exponential Golomb Encoder (EGE)

The term ‘sparse array’ is defined as an array where most of the samples are very small, and a few samples are large. Radar data (after the range FFT) is expected to be sparse in the range dimension. There are typically a few large samples corresponding to target reflections, the remaining samples are either the noise-floor or clutter or weak reflectors and are comparatively small. Most importantly, in a sparse array, the “average” bit-width (where bit-width is defined as the number of bits up to the most-significant 1) will be small.

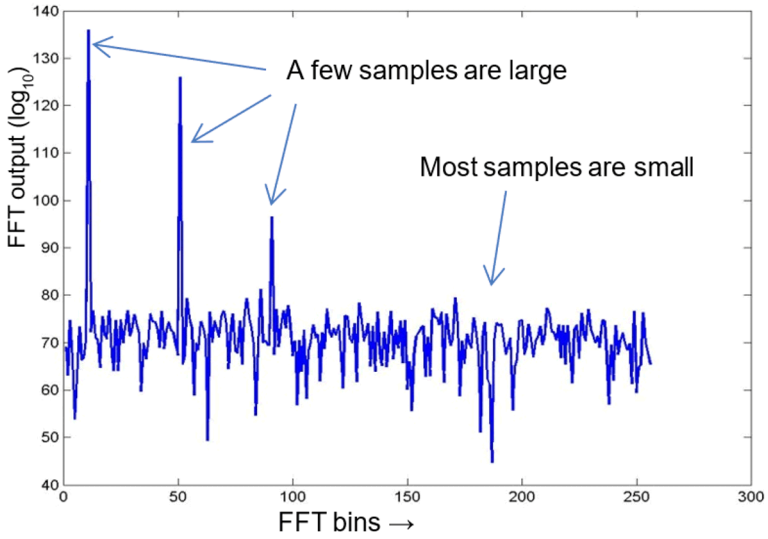


Figure 10-33. An example of sparse data

The “Order-k exponential Golomb encoder” (henceforth EGE) encodes each sample such that it occupies a space approximately proportional to its bit-width. A description of the algorithm is given in ‘https://en.wikipedia.org/wiki/Exponential-Golomb_coding’. Order-k Exponential Golomb codes are parameterized by the Golomb-parameter ‘’. This parameter represents the most common bitwidth in the input vector and is required to determine the boundary line between the variable-bitwidth quotient part (that is stored by having its length encoded in unary and the actual bits in binary form) and the fixed-bitwidth remainder part (that is stored in the usual binary form).

For example, if a number, say 23 were to be Exponential-Golomb encoded, then the process would look as follows.

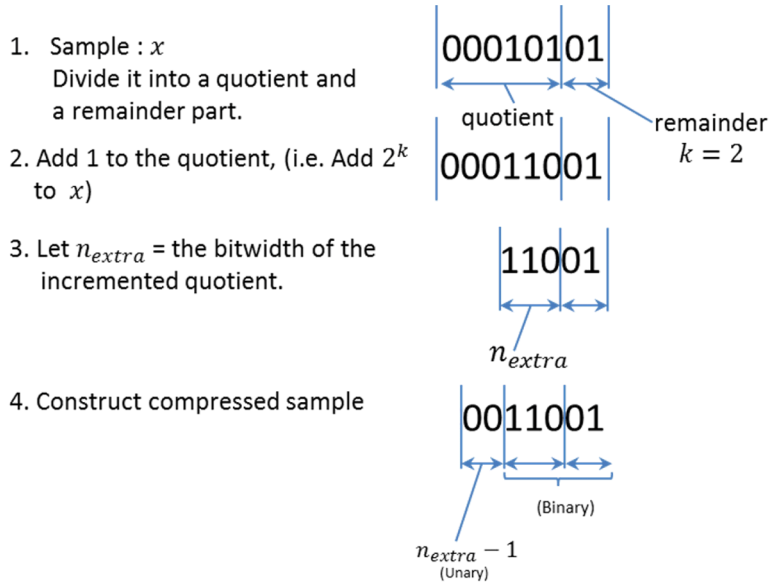


Figure 10-34. Exp-Golomb Encoding Example

One distinction in the compression engine is that the order of the EGE (i.e., Golomb parameter k) is automatically selected from a list of possible values (stored in an array called the 'Golomb parameter array') to optimize the Golomb parameter based on the input samples.

10.3.2.2.1 The EGE format

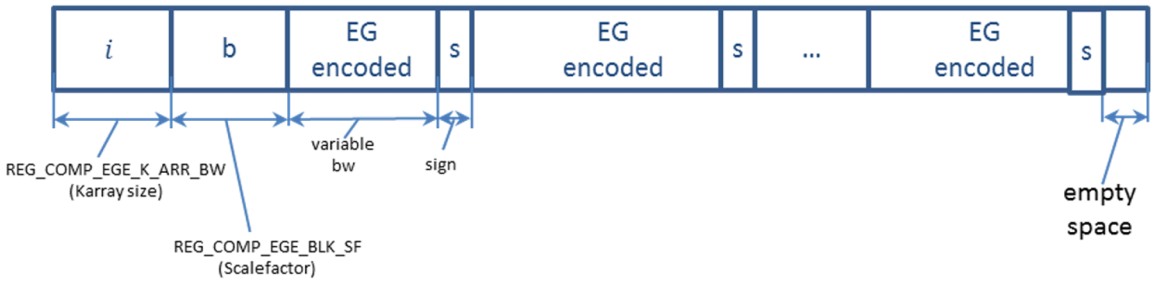


Figure 10-35. EGE Format

The block starts with a header, holding the scalefactor 'b', and the index to the Golomb parameter array 'i'. The rest of the block is filled with EGE words that take a variable number of bit-widths. Since the encoding is done only on the absolute value of each sample (i.e., without sign bit), the sign bit is taken separately, after each EGE word.

10.3.3 Operation

The compression/decompression operation is intended to be fairly invisible in operation. In order to use the compression/decompression engine the following steps are necessary.

- One 'param-set' of the HWA is configured to use the compression path. This configuration includes the number of samples per block, the access pattern, the number of blocks to be compressed, and some additional parameters for the compression engine.
- Before the compression engine is run, samples (real/complex) are to be placed in an input buffer of the HWA either as part of the operation of a previous param-set by the HWA or by the EDMA. Likewise before the decompression engine is run, compressed blocks are to be placed in an input buffer.
- The HWA is then executed. When the param-set corresponding to compression is reached the samples in the input buffer are read and compressed and then written to an output buffer as contiguous blocks. When the param-set corresponding to decompression is reached the samples in the output buffer are read, decompressed and then written to the output buffer.

Note : Internally, compression is accomplished via a two-pass operation. In the first pass, the samples are analyzed and the optimal parameters are selected. In the second pass, the samples are compressed to generate the compressed block of bits.

10.3.3.1 Configuring Compression

The following steps are to be followed to configure Compression

1. In order to configure the 'compression engine', set the compression path in the HWA by setting the ACCEL_MODE to 2, and then set the register CMP_DCMP to 0 to select the compression engine.
2. Enable dither (CMP_DITHER_ENABLE = 1), enable both first and second pass (CMP_PASS_SEL = 3), and enable the header (CMP_HEADER_EN = 1).
3. Configure the 'scale factor bitwidth'. For most cases, it should be set to the logarithm (in base 2) of number of bits per real sample. I.e. for e.g. CMP_SCALEFAC_BW = 4, if the sample bitwidth is 16bits (because), and CMP_SCALEFAC_BW = 5, if the sample bitwidth is 32 bits.
4. Configure BCNT to be 'number of blocks to be compressed'-1. For example, if there are 256 samples and the number of samples per block is 2, there would be 128 blocks. The BCNT register would then be configured to 127.
5. Setup the input formatter. In particular registers like SRCACNT, SRC_REAL, SRC_16b32b, and SRC_AINDX. SRCACNT should be set to the 'number of samples per block' – 1 (The '-1' in the previous equation comes from the fact that SRCACNT is zero based). SRC_AINDX would correspond to address increment after SRCACNT samples. For example, for compression of 4 complex 16 bit samples, the following configuration should be used.
   SRCACNT = 3 (4 samples)
   SRC_REAL = 0 (complex data)
   SRC_16b23b = 0 (16-bit samples)
6. Setup the output formatter. In particular registers like DSTACNT, DST_REAL, DST_16b32b, and DST_AINDX. DSTACNT should be set to the 'compressed data size (in samples)' – 1. (The '-1' in the previous equation comes from the fact that SRCACNT is zero based). For example, if a 50 % compression is required, DSTACNT can be set to ½ (SRCACNT+1)-1. To compress the previous example by 50 %, the following configuration should be used.
   DSTACNT = 1 (2 compressed samples)
   DST_REAL=1 (real data)
   DST_16b23b = 1 (32-bit samples)
7. Select/Configure the compression method. As of now, there is only a single method (EGE). (OR) As of now there are two compression algorithms (EGE and BFP).
   a. To configure EGE,
      i. Set the compression method (CMP_METHOD) to 0.
      ii. Program CMP_EGE_K_ARR_LEN which holds the length (in log2) of the list of golomb parameters and also CMP_EGE_K_ARR_<n> which holds the actual parameters.
      Point to note : The golomb parameter should correspond to the most common bit-width in the input array. Since radar data has a wide dynamic range, we typically set the golomb parameter list (for 16 bit numbers) to [0 2 4 6 8 10 12 15]. In the worst-case, the most common bitwidth can be as large as bitwidth of the input rhence the 15 at the end. In the case of 32-bit input, we set the list to [0 4 8 12 16 20 24 31]. CMP_EGE_K_ARR_LEN is normally set 3 (the list length is 8).
   b. (OR) To configure BFP,
      i. Set the compression method (CMP_METHOD) to 1.
      ii. Set the mantissa bitwidth - CMP_BFP_MANTISSA_BW
      The total compressed size of each block (in bits) is given by the following equation.
      if SRC_REAL == 1
      Compressed Size = CMP_SCALEFAC_BW + \
      (CMP_BFP_MANTISSA_BW x SRCACNT)
      elseif SRC_REAL == 0
      Compressed Size = CMP_SCALEFAC_BW + \
      (CMP_BFP_MANTISSA_BW x SRCACNT x 2)
      For example, for 50 % compression of 4 16-bit complex samples, the CMP_BFP_MANTISSA_BW should be set to 7, CMP_SCALEFAC_BW to 4. This is because there are 8 16-bit numbers

in the input (128 bits), and the compressed output should be less than 64 bits (including the header) i.e 64 CMP_SCALEFAC_BW + (CMP_BFP_MANTISSA_BW x SRCACNT x 2) Since CMP_SCALEFAC_BW = 4 and SRCACNT = 4, implies that CMP_BFP_MANTISSA_BW ≤ 7.

Points to note:

1. Configuring the input and output formatters: The sample size for the input formatter is dependent on SRC_REAL, and SRC_16b32b. If SRC_16b32b is set to 1 (i.e. 32 bits), then the per-real-sample bit-width is assumed to be 32bit (and 16bit otherwise). In each cycle, it will read 32 bits of data.
   The sample size of the output formatter is likewise dependent on DST_REAL and DST_16b32b. For example if DST_REAL=1, and DST_16b32b=1, then in each clock cycle the compression engine will write 32-bits to the output. If DST_REAL=0, and DST_16b32b=1, then in each cycle (subject to data availability) the compression engine will write 64 bits to the output buffer.
   The reason to care about the sample size in this case is that it directly limits the granularity of compression ratio. For example, consider the following scenario, assume 62.5 % compression is desired. In other words if the input is 128 bits (in total), the compressed output should be 80 bits (which is only divisible by 16 – and not by 32 or 64). So the output per-sample bit-width has to be configured to 16 which needs the destination to be configured to DST_REAL = 1, and DST_16b32b = 0.
2. Computing the actual compression Ratio.
   bits_per_src_sample = 16
   if SRC_16b32b == 1
   bits_per_src_sample = 32
   end
   if SRC_REAL == 0
   bits_per_src_sample = 2* bits_per_src_sample
   end
   Input_Block_Size = bits_per_src_sample * (SRC_ACNT + 1)
   bits_per_dst_sample = 16
   if DST_16b32b == 1
   bits_per_dst_sample = 32
   end
   if DST_REAL == 0
   bits_per_dst_sample = 2* bits_per_dst_sample
   end
   Output_Block_Size = bits_per_dst_sample * (DST_ACNT + 1)
   The compression ratio is then Output_Block_Size/Input_Block_Size.

10.3.3.2 Configuring Decompression

Decompression is configured almost exactly the same was as compression, except for the following differences.

- The only difference is that the register CMP_DCMP can be set to 1 to select the decompression engine
- The ‘input formatter configuration’ and the ‘output format configuration’ used in ‘the compression stage’ can be interchanged. For instance,
  - the SRCACNT used in compression can become the DSTACNT (and vice versa).
  - SRC_AINDX can be configured so as to jump to the next sample in the compressed block, whereas DST_AINDX can be configured to be the size of a sample .

10.3.3.3 Speed

When using either the BFP or EGE with 32-bit complex numbers, Compression and Decompression of one complex sample takes one cycle in steady state. If the sample consists of 64-bit complex numbers (i.e. 32-bit I, 32-bit Q), then EGE takes two cycles per sample in steady state, whereas BFP takes only a single cycle per complex number.

The initial delay (per paramSet) includes the cycles corresponding to the reading of the paramSet registers (~40 cycles) and an additional delay corresponding to the ‘number of samples in a block’.

10.3.3.4 Register Descriptions

10.3.3.4.1 Basic configuration

Register	Width	Parameter-Set?(Y/N)	Description
ACCEL_MODE	3	Y	The accelerator core is essentially a parallel set of paths. Each path performs a certain core operation, either FFT, CFAR-CA, compression/decompression, etc. To select the Compression/ Decompression Engine set to 2.
CMP_DCMP	1	Y	This register controls the compression mode, i.e. whether the operation to be performed is compression (when CMP_DCMP = 0) or decompression (when CMP_DCMP = 1).
CMP_METHOD	3	Y	3 bit register that selects the compression algorithm. The only valid value of the register is: Value: 0 — Description: EGE (Or) 3 bit register that selects one of the two compression algorithms. The only valid value of the register is: Value: 0 — Description: EGE Value: 1 — Description: BFP All other values are to be considered invalid

10.3.3.4.2 Compression/Decompression Configuration

Register	Width	Parameter-Set?(Y/N)	Description
CMP_DITHER_ENABLE	1	Y	The register enables dithering. Dithering prevents periodic quantization patterns from resulting in spurs. The dither source provides 3 bits of dither for every sample. Valid dither generation requires that the LFSR seed be programmed to any non-zero number. Note : • This register has to be set to 1 for proper operation of the hardware. accelerator. • While a separate LFSR is available to generate dither for compression, the same seed parameter is used by both FFT (when DITHER_TWID_EN is enabled) and compression. Hence, both LFSR_SEED and LFSR_LOAD have to be populated for the LFSR to generate valid dither.
CMP_PASS_SEL	2	Y	This register optionally bypasses the first pass (i.e. optimization of parameters) or the 2nd pass (actual compression). Note that if the first pass is bypassed, the programmed scale-factor is used. Value: 11 — Description: Both first pass and second pass are enabled. Value: 01 — Description: First pass is disabled.
CMP_HEADER_EN	1	Y	Optionally populate the header in the compressed stream. In most normal use-cases, this is set to 1, as the header is necessary for decompressing a block. However, if the first-pass is bypassed and all blocks are configured to use a specific customer-chosen scale-factor value, the header wouldn't be necessary.

Register	Width	Parameter-Set?(Y/N)	Description
CMP_SCALEFAC_BW	4	Y	The number of bits to be used in the header for the 'common scale-factor' per block. If the input is 16-bit (real or complex) set the common scale-factor to 4 (since the scale factor can vary from 0 to 15). If the input is 32-bit (real or complex), set the complex scale-factor to 5 (since the scale factor can vary from 0 to 31).

- Source/Destination configuration. Some of the Registers from the input and output formatters are reused internally to compute the compression ratio (with some additional description).

Register	Width	Parameter-Set? (Y/N)	Description
SRCACNT	12	Y	This register (plus 1) denotes the number of samples in a block. This is a zero-based count and therefore a register value of 15 indicates that there are 16 samples in a block. Note that we also rely on SRC_REAL and SRC_16b32b to denote the size of each sample, and they have to be correctly programmed for SRCACNT to select the necessary samples. Also, the maximum number of samples has to fit in the input buffer of the compression engine (< 2Kb).
DSTACNT	12	Y	This register (plus 1) denotes the desired output size in samples. To get the true compressed size in bits, DST_REAL, and DST_16b32b should be taken into consideration. When using the EG algorithm, the compression engine will compress all data so that it fits within this DSTACNT. (OR) When using the BFP algorithm, the programming of the mantissa bit-width, the header size and SRCACNT (i.e. the number of input samples) should be such that the compressed size is less than the DSTACNT.
BCNT	12	Y	This register (plus 1) denotes the number of blocks in the input buffer (ping and pong).
SRC_16b32b	1	Y	Specifies the number of bits samples per real sample. This register along with SRC_REAL, allows one to compute the number of bits per sample. Value: 0 — Description: 16 bit. Value: 1 — Description: 32 bit. Note : While BFP supports both 16 bit and 32 bit modes at the full rate, EGE only supports 16 bit at the full rate. 32 bit inputs will be processed at half rate.
SRC_REAL	1	Y	Specifies the format of the input samples Value: 0 — Description: REAL Value: 1 — Description: COMPLEX Note: The compression engine uses this register to process either 1 real sample per cycle or 1 complex sample per cycle. It is recommended that the COMPLEX mode is used to get the best throughput from the compression engine.

10.3.3.4.3 BFP Specific Registers

Register	Width	Parameter-Set?(Y/N)	Description
CMP_BFP_MANTISSA_BW	5	Y	The number of bits in the mantissa.
CMP_SCALEFAC	5	Y	Hardcoded scalefactor. This is used only if first pass is disabled (i.e. CMP_PASS_SEL = 01b

ADVANCE INFORMATION

10.3.3.4.4 EGE specific Registers

Register	Width	Parameter-Set?(Y/N)	Description
CMP_EGE_K_ARR_LEN	4	Y	This register value encodes the length of the 'list of Golomb parameters' to optimize over. The actual length of this list is 2^(Register Value). The valid range for this register is from 1 to 3. In effect the valid length of the list of parameters is 2, 4, 8.
CMP_EGE_K_ARR_<n>	5bits per element (upto 8 elements in total)	N	These set of registers hold the list of golomb parameters to optimize over. The number of valid elements is determined by the CMP_EGE_K_ARR_LEN (see register).

10.3.3.4.5 Parameter Set

The parameter sets for all devices that have the compression engine present are show below.

Figure 10-36. Compression engine - parameter-set arrangement (xWRL6432, xWRL1432)

10.4 HWA_CFG Registers

Table 10-15 lists the memory-mapped registers for the HWA_CFG registers. All register offset addresses not listed in Table 10-15 should be considered as reserved locations and the register contents should not be modified.

Table 10-15. HWA_CFG Registers

Offset	Acronym	Register Name	Section
0h	HWACCREG1		Go
4h	HWACCREG2		Go
8h	HWACCREG3		Go
Ch	HWACCREG4		Go
10h	HWACCREG5		Go
14h	HWACCREG6		Go
18h	HWACCREG7		Go
1Ch	HWACCREG8		Go
20h	HWACCREG11		Go
24h	HWACCREG12		Go
28h	HWACCREG13		Go
2Ch	HWACCREG14		Go
30h	HWACCREG15		Go
34h	CFAR_DET_THR		Go
38h	MAX1VALUE		Go
3Ch	MAX1INDEX		Go
40h	ISUM1LSB		Go
44h	ISUM1MSB		Go
48h	QSUM1LSB		Go
4Ch	QSUM1MSB		Go
50h	MAX2VALUE		Go
54h	MAX2INDEX		Go
58h	ISUM2LSB		Go
5Ch	ISUM2MSB		Go
60h	QSUM2LSB		Go
64h	QSUM2MSB		Go
68h	MAX3VALUE		Go
6Ch	MAX3INDEX		Go
70h	ISUM3LSB		Go
74h	ISUM3MSB		Go
78h	QSUM3LSB		Go
7Ch	QSUM3MSB		Go
80h	MAX4VALUE		Go
84h	MAX4INDEX		Go
88h	ISUM4LSB		Go
8Ch	ISUM4MSB		Go
90h	QSUM4LSB		Go
94h	QSUM4MSB		Go
98h	CFARTEST		Go
9Ch	RDSTATUS		Go
A0h	SIGDMACH1DONE		Go
A4h	SIGDMACH2DONE		Go

Table 10-15. HWA_CFG Registers (continued)

Offset	Acronym	Register Name	Section
A8h	SIGDMACH3DONE		Go
ACh	SIGDMACH4DONE		Go
B0h	SIGDMACH5DONE		Go
B4h	SIGDMACH6DONE		Go
B8h	SIGDMACH7DONE		Go
BCh	SIGDMACH8DONE		Go
C0h	SIGDMACH9DONE		Go
C4h	SIGDMACH10DONE		Go
C8h	SIGDMACH11DONE		Go
CCh	SIGDMACH12DONE		Go
D0h	SIGDMACH13DONE		Go
D4h	SIGDMACH14DONE		Go
D8h	SIGDMACH15DONE		Go
DCh	SIGDMACH16DONE		Go
E0h	MEMACCESSERR		Go
E4h	FFTCLIP		Go
E8h	FFTPEAKCNT		Go
ECh	HWACCREG1RD		Go
F0h	HWACCREG2RD		Go
F4h	HWACCREG3RD		Go
F8h	CMP_EGE_K0123		Go
FCh	CMP_EGE_K4567		Go
100h	HWA_SAFETY_ENABLE		Go
104h	MEMINIT		Go
108h	MEMINITDONE		Go
10Ch	HWA_SAFETY_WIN_RAM_ERR_LOC		Go
110h	HWA_SAFETY_PARAM_RAM_ERR_LOC		Go
114h	HWA_SAFETY_IPING_ERR_LOC		Go
118h	HWA_SAFETY_IPONG_ERR_LOC		Go
11Ch	HWA_SAFETY_OPING_ERR_LOC		Go
120h	HWA_SAFETY_OPONG_ERR_LOC		Go
124h	FFTINTMEMWRDATA		Go
128h	FFTINTMEMRDDATA		Go
12Ch	HWACCREG16		Go
130h	DCEST1I_SW		Go
134h	DCEST2I_SW		Go
138h	DCEST3I_SW		Go
13Ch	DCEST4I_SW		Go
140h	DCEST5I_SW		Go
144h	DCEST6I_SW		Go
148h	DCEST1I		Go
14Ch	DCEST2I		Go
150h	DCEST3I		Go
154h	DCEST4I		Go
158h	DCEST5I		Go
15Ch	DCEST6I		Go
160h	DC_ACC1I_LSB		Go

Table 10-15. HWA_CFG Registers (continued)

Offset	Acronym	Register Name	Section
164h	DC_ACC1I_MSB		Go
168h	DC_ACC2I_LSB		Go
16Ch	DC_ACC2I_MSB		Go
170h	DC_ACC3I_LSB		Go
174h	DC_ACC3I_MSB		Go
178h	DC_ACC4I_LSB		Go
17Ch	DC_ACC4I_MSB		Go
180h	DC_ACC5I_LSB		Go
184h	DC_ACC5I_MSB		Go
188h	DC_ACC6I_LSB		Go
18Ch	DC_ACC6I_MSB		Go
190h	DCEST1Q_SW		Go
194h	DCEST2Q_SW		Go
198h	DCEST3Q_SW		Go
19Ch	DCEST4Q_SW		Go
1A0h	DCEST5Q_SW		Go
1A4h	DCEST6Q_SW		Go
1A8h	DCEST1Q		Go
1ACh	DCEST2Q		Go
1B0h	DCEST3Q		Go
1B4h	DCEST4Q		Go
1B8h	DCEST5Q		Go
1BCh	DCEST6Q		Go
1C0h	DC_ACC1Q_LSB		Go
1C4h	DC_ACC1Q_MSB		Go
1C8h	DC_ACC2Q_LSB		Go
1CCh	DC_ACC2Q_MSB		Go
1D0h	DC_ACC3Q_LSB		Go
1D4h	DC_ACC3Q_MSB		Go
1D8h	DC_ACC4Q_LSB		Go
1DCh	DC_ACC4Q_MSB		Go
1E0h	DC_ACC5Q_LSB		Go
1E4h	DC_ACC5Q_MSB		Go
1E8h	DC_ACC6Q_LSB		Go
1ECh	DC_ACC6Q_MSB		Go
1F0h	DCACC1_CLIP		Go
1F4h	DCACC2_CLIP		Go
1F8h	DCACC3_CLIP		Go
1FCh	DCACC4_CLIP		Go
200h	DCACC5_CLIP		Go
204h	DCACC6_CLIP		Go
208h	DCEST1_CLIP		Go
20Ch	DCEST2_CLIP		Go
210h	DCEST3_CLIP		Go
214h	DCEST4_CLIP		Go
218h	DCEST5_CLIP		Go
21Ch	DCEST6_CLIP		Go

Table 10-15. HWA_CFG Registers (continued)

Offset	Acronym	Register Name	Section
220h	DCSUB1_CLIP		Go
224h	DCSUB2_CLIP		Go
228h	DCSUB3_CLIP		Go
22Ch	DCSUB4_CLIP		Go
230h	DCSUB5_CLIP		Go
234h	DCSUB6_CLIP		Go
238h	DCEST_SHIFT		Go
23Ch	DCEST_SCALE		Go
240h	INTF_MAG_SCALE		Go
244h	INTF_MAG_SHIFT		Go
248h	INTF_MAGDIFF_SCALE		Go
24Ch	INTF_MAGDIFF_SHIFT		Go
250h	INTF_FRAME_ZEROCOUNT		Go
254h	INTF_CHIRP_ZEROCOUNT		Go
258h	INTF_MAGTHRESH1_SW		Go
25Ch	INTF_MAGTHRESH2_SW		Go
260h	INTF_MAGTHRESH3_SW		Go
264h	INTF_MAGTHRESH4_SW		Go
268h	INTF_MAGTHRESH5_SW		Go
26Ch	INTF_MAGTHRESH6_SW		Go
270h	INTF_MAGDIFFTHRESH1_SW		Go
274h	INTF_MAGDIFFTHRESH2_SW		Go
278h	INTF_MAGDIFFTHRESH3_SW		Go
27Ch	INTF_MAGDIFFTHRESH4_SW		Go
280h	INTF_MAGDIFFTHRESH5_SW		Go
284h	INTF_MAGDIFFTHRESH6_SW		Go
288h	INTF_MAGACC1_LSB		Go
28Ch	INTF_MAGACC1_MSB		Go
290h	INTF_MAGACC2_LSB		Go
294h	INTF_MAGACC2_MSB		Go
298h	INTF_MAGACC3_LSB		Go
29Ch	INTF_MAGACC3_MSB		Go
2A0h	INTF_MAGACC4_LSB		Go
2A4h	INTF_MAGACC4_MSB		Go
2A8h	INTF_MAGACC5_LSB		Go
2ACh	INTF_MAGACC5_MSB		Go
2B0h	INTF_MAGACC6_LSB		Go
2B4h	INTF_MAGACC6_MSB		Go
2B8h	INTF_MAGDIFFACC1_LSB		Go
2BCh	INTF_MAGDIFFACC1_MSB		Go
2C0h	INTF_MAGDIFFACC2_LSB		Go
2C4h	INTF_MAGDIFFACC2_MSB		Go
2C8h	INTF_MAGDIFFACC3_LSB		Go
2CCh	INTF_MAGDIFFACC3_MSB		Go
2D0h	INTF_MAGDIFFACC4_LSB		Go
2D4h	INTF_MAGDIFFACC4_MSB		Go
2D8h	INTF_MAGDIFFACC5_LSB		Go

Table 10-15. HWA_CFG Registers (continued)

Offset	Acronym	Register Name	Section
2DCh	INTF_MAGDIFFACC5_MSB		Go
2E0h	INTF_MAGDIFFACC6_LSB		Go
2E4h	INTF_MAGDIFFACC6_MSB		Go
2E8h	INTF_MAGACC1_CLIP		Go
2ECh	INTF_MAGACC2_CLIP		Go
2F0h	INTF_MAGACC3_CLIP		Go
2F4h	INTF_MAGACC4_CLIP		Go
2F8h	INTF_MAGACC5_CLIP		Go
2FCh	INTF_MAGACC6_CLIP		Go
300h	INTF_MAGDIFFACC1_CLIP		Go
304h	INTF_MAGDIFFACC2_CLIP		Go
308h	INTF_MAGDIFFACC3_CLIP		Go
30Ch	INTF_MAGDIFFACC4_CLIP		Go
310h	INTF_MAGDIFFACC5_CLIP		Go
314h	INTF_MAGDIFFACC6_CLIP		Go
318h	INTF_MAGTHRESH1		Go
31Ch	INTF_MAGTHRESH2		Go
320h	INTF_MAGTHRESH3		Go
324h	INTF_MAGTHRESH4		Go
328h	INTF_MAGTHRESH5		Go
32Ch	INTF_MAGTHRESH6		Go
330h	INTF_MAGDIFFTHRESH1		Go
334h	INTF_MAGDIFFTHRESH2		Go
338h	INTF_MAGDIFFTHRESH3		Go
33Ch	INTF_MAGDIFFTHRESH4		Go
340h	INTF_MAGDIFFTHRESH5		Go
344h	INTF_MAGDIFFTHRESH6		Go
348h	INTF_SUMMAGTHRESH		Go
34Ch	INTF_SUMMAGDIFFTHRESH		Go
350h	INTF_SUMMAGTHRESH_CLIP		Go
354h	INTF_SUMMAGDIFFTHRESH_CLIP		Go
358h	CMULTSCALE1I		Go
35Ch	CMULTSCALE2I		Go
360h	CMULTSCALE3I		Go
364h	CMULTSCALE4I		Go
368h	CMULTSCALE5I		Go
36Ch	CMULTSCALE6I		Go
370h	CMULTSCALE1Q		Go
374h	CMULTSCALE2Q		Go
378h	CMULTSCALE3Q		Go
37Ch	CMULTSCALE4Q		Go
380h	CMULTSCALE5Q		Go
384h	CMULTSCALE6Q		Go
388h	CLR_MISC_CLIP		Go
38Ch	FFTINTMEMADDR		Go
390h	INTF_STATS_RESET_SW		Go
394h	DCEST_RESET_SW		Go

Table 10-15. HWA_CFG Registers (continued)

Offset	Acronym	Register Name	Section
398h	IP_OP_FORMATTER_CLIP_STATUS		Go
39Ch	INTF_MAGTHRESH1_CLIP		Go
3A0h	INTF_MAGTHRESH2_CLIP		Go
3A4h	INTF_MAGTHRESH3_CLIP		Go
3A8h	INTF_MAGTHRESH4_CLIP		Go
3ACh	INTF_MAGTHRESH5_CLIP		Go
3B0h	INTF_MAGTHRESH6_CLIP		Go
3B4h	INTF_MAGDIFFTHRESH1_CLIP		Go
3B8h	INTF_MAGDIFFTHRESH2_CLIP		Go
3BCh	INTF_MAGDIFFTHRESH3_CLIP		Go
3C0h	INTF_MAGDIFFTHRESH4_CLIP		Go
3C4h	INTF_MAGDIFFTHRESH5_CLIP		Go
3C8h	INTF_MAGDIFFTHRESH6_CLIP		Go
3CCh	HWA_SAFETY_ERR_MASK		Go
3D0h	HWA_SAFETY_ERR_STATUS		Go
3D4h	HWA_SAFETY_ERR_STATUS_RAW		Go

Complex bit access types are encoded to fit into small table cells. Table 10-16 shows the codes that are used for access types in this section.

Table 10-16. HWA_CFG 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

10.4.1 HWACCREG1 Register (Offset = 0h) [Reset = 00000000h]

HWACCREG1 is shown in Table 10-17.

Return to the Summary Table.

Table 10-17. HWACCREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	NU2	R	0h	
12	ACCDYNCLKEN_LEVEL2	R/W	0h	Level 2 dynamic clock-gating control :- Setting this register bit to 1 will lead to further power saving by disabling clock during FSM wait state.
11	ACCDYNCLKEN	R/W	0h	Dynamic Clock-gating Control:Setting this register bit to 1 enables the capability to clock gate the Radar Accelerator core IPs (FFT and CFAR-CA datapath,CFAR-OS datapath, memory compression datapath) based on the ParamSet being executed.

Table 10-17. HWACCREG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	FFT1DEN	R/W	0h	ADC buffer sharing mode This register is relevant where the Radar Hardware Accelerator is included in a single device along with the mmWave RF front-end. In such a case, during active chirp transmission and inline 1st dimension FFT processing, the ACCEL_MEM0 and ACCEL_MEM1 memories of the accelerator are shared as ping-pong ADC buffers. This register bit needs to be set during this time, so that while the Digital Front End writes ADC samples to the ping buffer, the accelerator automatically accesses (only) the pong buffer, and vice versa. At the end of the active transmission portion of a frame, this bit can be cleared, so that the accelerator has access to all the four local memories independently.
9	NU1	R	0h	
8-6	ACCRESET	R/W	0h	Software Reset Control: This register provides software reset control for the Radar Hardware Accelerator. The assertion of these register bits by the main processor will bring the Accelerator Engine to a known reset state. This is mostly applicable for resetting the accelerator in case of unexpected behavior. The sequence to be followed in case software reset is to write 111b to this register and then a 000b
5-3	ACCCLKEN	R/W	0h	Clock-gating Control: This register bit controls the enable/disable for the clock of the Radar Accelerator. This register bit can be set to 0 to clock-gate the accelerator when not using the accelerator. Before enabling the accelerator or before configuring the registers of accelerator, this register bit should be set to 111b first, so that the clock is available.
2-0	ACCENABLE	R/W	0h	Enable/Disable Control: A value of ACC_ENABLE = 111b enables the Radar Hardware Accelerator and any other value of the register keeps the Accelerator Engine in disabled state.

10.4.2 HWACCREG2 Register (Offset = 4h) [Reset = 00000000h]

HWACCREG2 is shown in Table 10-18.

Return to the Summary Table.

Table 10-18. HWACCREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	
15-0	DMA2ACCTRIG		0h	DMA trigger register: This register is relevant whenever DMA triggered mode is used (i.e., TRIGMODE = 011b). Whenever a DMA channel has finished copying input samples into the local memory of the accelerator and wants to trigger the accelerator, the procedure to follow is to use a second linked DMA channel to write a 16-bit one-hot signature into this register to trigger the accelerator. In DMA triggered mode, the State Machine keeps monitoring this 16-bit register and waits as long as a specific bit (see DMA2ACC_CHANNEL_TRIGSRC) in this register is zero. The second linked DMA channel writes a one-hot signature that sets the specific bit, so that the State Machine gets triggered and starts the accelerator operations for that parameter-set.

10.4.3 HWACCREG3 Register (Offset = 8h) [Reset = 00000000h]

HWACCREG3 is shown in Table 10-19.

Return to the Summary Table.

Table 10-19. HWACCREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CM42DMATRIG		0h	Override accelerator Trigger to DMA.Can be used for triggering the first and second DMA transfer thorugh processor
15-1	NU	R	0h	
0	CM42ACCTRIG		0h	Software trigger bit: This register bit is relevant whenever software triggered mode is used (i.e., TRIGMODE = 001b). The main processor software can set this register bit, so that the State Machine gets triggered and starts the accelerator operations for that parameter-set.

10.4.4 HWACCREG4 Register (Offset = Ch) [Reset = 00000000h]

HWACCREG4 is shown in Table 10-20.

Return to the Summary Table.

Table 10-20. HWACCREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R/W	0h	Spare register

10.4.5 HWACCREG5 Register (Offset = 10h) [Reset = 00000000h]

HWACCREG5 is shown in Table 10-21.

Return to the Summary Table.

Table 10-21. HWACCREG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BPMPATTERNMSB	R/W	0h	BPM pattern MSB: Specifies the BPM pattern to be used to multiply the input samples if BPM removal is enabled

10.4.6 HWACCREG6 Register (Offset = 14h) [Reset = 00000000h]

HWACCREG6 is shown in Table 10-22.

Return to the Summary Table.

Table 10-22. HWACCREG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BPMPATTERNLSB	R/W	0h	BPM pattern LSB: Specifies the BPM pattern to be used to multiply the input samples if BPM removal is enabled

10.4.7 HWACCREG7 Register (Offset = 18h) [Reset = 00000000h]

HWACCREG7 is shown in Table 10-23.

Return to the Summary Table.

Table 10-23. HWACCREG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU3	R	0h	

Table 10-23. HWACCREG7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	STG1LUTSELWR	R/W	0h	Select Window RAM or Internal RAM: The Internal RAM for Vector Multiplication mode is mapped to the same address space as the Window RAM. Hence, this register bit is required to specify which of these two needs to be selected, when loading the co-efficients via DMA or M4. 0 - Window RAM is selected 1 - Internal RAM for Vector Multiplication mode is selected. Keep this register bit as 0 always, except during the period when Internal RAM needs to be loaded.
23-17	NU2	R	0h	
16	DITHERTWIDEN	R/W	0h	Twiddle factor dithering enable: This register-bit is used to enable/ disable dithering of twiddle factors in the FFT.
15-10	NU1	R	0h	
9-0	BPMRATE	R/W	0h	BPM rate: Specifies the number of input samples corresponding to each BPM bit. Minimum valid value for this register is 1.

10.4.8 HWACCREG8 Register (Offset = 1Ch) [Reset = 00000000h]

HWACCREG8 is shown in Table 10-24.

Return to the Summary Table.

Table 10-24. HWACCREG8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU2	R	0h	
28-24	FFTSUMDIV	R/W	0h	Right-shifting for Sum Statistic: This register specifies how many LSBs to drop to convert the sum statistics to 24-bit value going to the Output Formatter
23-0	NU1	R	0h	

10.4.9 HWACCREG11 Register (Offset = 20h) [Reset = 00000000h]

HWACCREG11 is shown in Table 10-25.

Return to the Summary Table.

Table 10-25. HWACCREG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	LFSRLOAD		0h	To load the LFSR seed, a pulse signal needs to be provided, by writing a 1 to the LFSR_LOAD register-bit. Self clearing
30-29	NU	R	0h	
28-0	LFSRSEED	R/W	0h	LFSR seed value (random pattern) for twiddle factor dithering,

10.4.10 HWACCREG12 Register (Offset = 24h) [Reset = 00000000h]

HWACCREG12 is shown in Table 10-26.

Return to the Summary Table.

Table 10-26. HWACCREG12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU2	R	0h	

Table 10-26. HWACCREG12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	ACC_TRIGGER_IN_CLR		0h	Clear trigger status read-only register: This register-bit when set clears the trigger status register ACC_TRIG_IN_STAT described above
23-19	NU1	R	0h	
18-0	ACC_TRIGGER_IN_STAT	R	0h	Debug register for trigger status: This is a read-only status register, which indicates the trigger status of the accelerator, i.e., whether a specific DMA trigger or a Ping-pong trigger or a SW trigger was ever received (refer TRIGMODE in HW_ACC_PARAM register set). The MSB 16 bits of this register indicate whether a trigger was received via DMA trigger method. The next two bits (i.e., bit indices 2 and 1) indicate the status of DFE ping-pong switch-based trigger and SW trigger respectively. The LSB bit is always 1 and can be ignored {DMA2ACCTRIG [15:0],adc_buffer_done,CM42ACCTRIG,1}

10.4.11 HWACCREG13 Register (Offset = 28h) [Reset = 00000000h]

HWACCREG13 is shown in Table 10-27.

Return to the Summary Table.

Table 10-27. HWACCREG13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	NU	R	0h	
17-0	CFAR_THRESH	R/W	0h	CFAR Threshold scale factor: This value is used to either multiply or add to the surrounding noise average to determine the threshold used for detection of the present cell under test. If logarithmic CFAR mode is disabled (i.e., in magnitude or magnitude-squared mode), then the register value is multiplied with the surrounding noise average to determine the threshold, else it is added to the surrounding noise average. In the former case, this 18-bit register is interpreted as a 14.4 value. In the latter case (i.e., logarithmic mode), the 18-bit register is interpreted as a 7.11 value.

10.4.12 HWACCREG14 Register (Offset = 2Ch) [Reset = 00000000h]

HWACCREG14 is shown in Table 10-28.

Return to the Summary Table.

Table 10-28. HWACCREG14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PARAMDONESTAT	R	0h	Parameter-set done status: This read-only status register can be used by the main processor to see which parameter-sets are complete that led to the interrupt to the main processor. The individual bits in this 32-bit status register indicate which of the 32 parameter-sets have completed.

10.4.13 HWACCREG15 Register (Offset = 30h) [Reset = 00000000h]

HWACCREG15 is shown in Table 10-29.

Return to the Summary Table.

Table 10-29. HWACCREG15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PARAMDONECLR		0h	Status bits in PARAMDONESTAT are not automatically cleared, but they can be individually cleared by writing to 32-bit register PARAMDONECLR.

10.4.14 CFAR_DET_THR Register (Offset = 34h) [Reset = 00000000h]

CFAR_DET_THR is shown in Table 10-30.

Return to the Summary Table.

Table 10-30. CFAR_DET_THR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU	R	0h	
23-0	CFAR_DET_THR	R/W	0h	This register is used to specify the threshold used for the detection of the present cell under test during CFAR-CA mode when number of samples for left side and right side noise averaging is 0.

10.4.15 MAX1VALUE Register (Offset = 38h) [Reset = 00000000h]

MAX1VALUE is shown in Table 10-31.

Return to the Summary Table.

Table 10-31. MAX1VALUE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU	R	0h	
23-0	MAX1VALUE	R	0h	Max value: These registers contain the max value on a per-iteration basis. These registers are meaningful only when Magnitude or Log-Magnitude is enabled. Only the max values for up to four iterations are recorded in these registers. For larger number of iterations, use Statistics output mode (FFT_OUT_MODE in HW_ACC_PARAM register set).

10.4.16 MAX1INDEX Register (Offset = 3Ch) [Reset = 00000000h]

MAX1INDEX is shown in Table 10-32.

Return to the Summary Table.

Table 10-32. MAX1INDEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11-0	MAX1INDEX	R	0h	Max index: These registers contain the max index on a per-iteration basis, corresponding to each max value in the MAXn_VALUE registers.

10.4.17 ISUM1LSB Register (Offset = 40h) [Reset = 00000000h]

ISUM1LSB is shown in Table 10-33.

Return to the Summary Table.

Table 10-33. ISUM1LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ISUM1LSB	R	0h	Sum statistics: These registers contain the sum of the I outputs and Q outputs on a per-iteration basis. Only the statistics for up to four iterations are recorded in these registers. For larger number of iterations, use Statistics output mode (FFT_OUT_MODE in HW_ACC_PARAM register set).

10.4.18 ISUM1MSB Register (Offset = 44h) [Reset = 00000000h]

ISUM1MSB is shown in Table 10-34.

Return to the Summary Table.

Table 10-34. ISUM1MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	ISUM1MSB	R	0h	Refer ISUM1LSB

10.4.19 QSUM1LSB Register (Offset = 48h) [Reset = 00000000h]

QSUM1LSB is shown in Table 10-35.

Return to the Summary Table.

Table 10-35. QSUM1LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QSUM1LSB	R	0h	Refer ISUM1LSB

10.4.20 QSUM1MSB Register (Offset = 4Ch) [Reset = 00000000h]

QSUM1MSB is shown in Table 10-36.

Return to the Summary Table.

Table 10-36. QSUM1MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	QSUM1MSB	R	0h	Refer ISUM1LSB

10.4.21 MAX2VALUE Register (Offset = 50h) [Reset = 00000000h]

MAX2VALUE is shown in Table 10-37.

Return to the Summary Table.

Table 10-37. MAX2VALUE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU	R	0h	
23-0	MAX2VALUE	R	0h	Refer MAX1VALUE

10.4.22 MAX2INDEX Register (Offset = 54h) [Reset = 00000000h]

MAX2INDEX is shown in Table 10-38.

Return to the Summary Table.

Table 10-38. MAX2INDEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11-0	MAX2INDEX	R	0h	Refer MAX1INDEX

10.4.23 ISUM2LSB Register (Offset = 58h) [Reset = 00000000h]

ISUM2LSB is shown in Table 10-39.

Return to the Summary Table.

Table 10-39. ISUM2LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ISUM2LSB	R	0h	Refer ISUM1LSB

10.4.24 ISUM2MSB Register (Offset = 5Ch) [Reset = 00000000h]

ISUM2MSB is shown in Table 10-40.

Return to the Summary Table.

Table 10-40. ISUM2MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	ISUM2MSB	R	0h	Refer ISUM1LSB

10.4.25 QSUM2LSB Register (Offset = 60h) [Reset = 00000000h]

QSUM2LSB is shown in Table 10-41.

Return to the Summary Table.

Table 10-41. QSUM2LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QSUM2LSB	R	0h	Refer ISUM1LSB

10.4.26 QSUM2MSB Register (Offset = 64h) [Reset = 00000000h]

QSUM2MSB is shown in Table 10-42.

Return to the Summary Table.

Table 10-42. QSUM2MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	QSUM2MSB	R	0h	Refer ISUM1LSB

10.4.27 MAX3VALUE Register (Offset = 68h) [Reset = 00000000h]

MAX3VALUE is shown in Table 10-43.

Return to the Summary Table.

Table 10-43. MAX3VALUE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU	R	0h	

Table 10-43. MAX3VALUE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-0	MAX3VALUE	R	0h	Refer MAX1VALUE

10.4.28 MAX3INDEX Register (Offset = 6Ch) [Reset = 00000000h]

MAX3INDEX is shown in Table 10-44.

Return to the Summary Table.

Table 10-44. MAX3INDEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11-0	MAX3INDEX	R	0h	Refer MAX1INDEX

10.4.29 ISUM3LSB Register (Offset = 70h) [Reset = 00000000h]

ISUM3LSB is shown in Table 10-45.

Return to the Summary Table.

Table 10-45. ISUM3LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ISUM3LSB	R	0h	Refer ISUM1LSB

10.4.30 ISUM3MSB Register (Offset = 74h) [Reset = 00000000h]

ISUM3MSB is shown in Table 10-46.

Return to the Summary Table.

Table 10-46. ISUM3MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	ISUM3MSB	R	0h	Refer ISUM1LSB

10.4.31 QSUM3LSB Register (Offset = 78h) [Reset = 00000000h]

QSUM3LSB is shown in Table 10-47.

Return to the Summary Table.

Table 10-47. QSUM3LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QSUM3LSB	R	0h	Refer ISUM1LSB

10.4.32 QSUM3MSB Register (Offset = 7Ch) [Reset = 00000000h]

QSUM3MSB is shown in Table 10-48.

Return to the Summary Table.

Table 10-48. QSUM3MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	QSUM3MSB	R	0h	Refer ISUM1LSB

10.4.33 MAX4VALUE Register (Offset = 80h) [Reset = 00000000h]

MAX4VALUE is shown in Table 10-49.

Return to the Summary Table.

Table 10-49. MAX4VALUE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU	R	0h	
23-0	MAX4VALUE	R	0h	Refer MAX1INDEX

10.4.34 MAX4INDEX Register (Offset = 84h) [Reset = 00000000h]

MAX4INDEX is shown in Table 10-50.

Return to the Summary Table.

Table 10-50. MAX4INDEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11-0	MAX4INDEX	R	0h	Refer MAX1VALUE

10.4.35 ISUM4LSB Register (Offset = 88h) [Reset = 00000000h]

ISUM4LSB is shown in Table 10-51.

Return to the Summary Table.

Table 10-51. ISUM4LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ISUM4LSB	R	0h	Refer ISUM1LSB

10.4.36 ISUM4MSB Register (Offset = 8Ch) [Reset = 00000000h]

ISUM4MSB is shown in Table 10-52.

Return to the Summary Table.

Table 10-52. ISUM4MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	ISUM4MSB	R	0h	Refer ISUM1LSB

10.4.37 QSUM4LSB Register (Offset = 90h) [Reset = 00000000h]

QSUM4LSB is shown in Table 10-53.

Return to the Summary Table.

Table 10-53. QSUM4LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QSUM4LSB	R	0h	Refer ISUM1LSB

10.4.38 QSUM4MSB Register (Offset = 94h) [Reset = 00000000h]

QSUM4MSB is shown in Table 10-54.

Return to the Summary Table.

Table 10-54. QSUM4MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU	R	0h	
3-0	QSUM4MSB	R	0h	Refer ISUM1LSB

10.4.39 CFARTEST Register (Offset = 98h) [Reset = 00000000h]

CFARTEST is shown in Table 10-55.

Return to the Summary Table.

Table 10-55. CFARTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU	R	0h	
23-0	CFARTEST	R/W	0h	Reserved.TI internal

10.4.40 RDSTATUS Register (Offset = 9Ch) [Reset = 00000000h]

RDSTATUS is shown in Table 10-56.

Return to the Summary Table.

Table 10-56. RDSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU	R	0h	
16-5	LOOPCNT	R	0h	Running value of the loop count when the HWA is executing from PARAM RAM . For Debug only
4-0	PARAMADDR	R	0h	Index of the current parameter set being executed from PARAM RAM . For Debug only

10.4.41 SIGDMACH1DONE Register (Offset = A0h) [Reset = 00000001h]

SIGDMACH1DONE is shown in Table 10-57.

Return to the Summary Table.

Table 10-57. SIGDMACH1DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH1DONE	R	1h	Signature for DMA channel 1 completion (tied to 0x0001 in HW). Linked DMA can copy from one of these SIG_DMACHx_DONE registers into DMA2ACC_TRIG register to set the appropriate register bit to signal the completion of DMA and trigger the accelerator

10.4.42 SIGDMACH2DONE Register (Offset = A4h) [Reset = 00000002h]

SIGDMACH2DONE is shown in Table 10-58.

Return to the Summary Table.

Table 10-58. SIGDMACH2DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH2DONE	R	2h	Signature for DMA channel 2 completion (tied to 0x0002 in HW)

10.4.43 SIGDMACH3DONE Register (Offset = A8h) [Reset = 00000004h]

SIGDMACH3DONE is shown in Table 10-59.

Return to the Summary Table.

Table 10-59. SIGDMACH3DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH3DONE	R	4h	Signature for DMA channel 3 completion (tied to 0x0004 in HW)

10.4.44 SIGDMACH4DONE Register (Offset = ACh) [Reset = 00000008h]

SIGDMACH4DONE is shown in Table 10-60.

Return to the Summary Table.

Table 10-60. SIGDMACH4DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH4DONE	R	8h	Signature for DMA channel 4 completion (tied to 0x0008 in HW)

10.4.45 SIGDMACH5DONE Register (Offset = B0h) [Reset = 00000010h]

SIGDMACH5DONE is shown in Table 10-61.

Return to the Summary Table.

Table 10-61. SIGDMACH5DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH5DONE	R	10h	Signature for DMA channel 5 completion (tied to 0x0010 in HW)

10.4.46 SIGDMACH6DONE Register (Offset = B4h) [Reset = 00000020h]

SIGDMACH6DONE is shown in Table 10-62.

Return to the Summary Table.

Table 10-62. SIGDMACH6DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH6DONE	R	20h	Signature for DMA channel 6 completion (tied to 0x0020 in HW)

10.4.47 SIGDMACH7DONE Register (Offset = B8h) [Reset = 00000040h]

SIGDMACH7DONE is shown in Table 10-63.

Return to the Summary Table.

Table 10-63. SIGDMACH7DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH7DONE	R	40h	Signature for DMA channel 7 completion (tied to 0x0040 in HW)

10.4.48 SIGDMACH8DONE Register (Offset = BCh) [Reset = 00000080h]

SIGDMACH8DONE is shown in Table 10-64.

Return to the Summary Table.

Table 10-64. SIGDMACH8DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH8DONE	R	80h	Signature for DMA channel 8 completion (tied to 0x0080 in HW)

10.4.49 SIGDMACH9DONE Register (Offset = C0h) [Reset = 00000100h]

SIGDMACH9DONE is shown in Table 10-65.

Return to the Summary Table.

Table 10-65. SIGDMACH9DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH9DONE	R	100h	Signature for DMA channel 9 completion (tied to 0x0100 in HW)

10.4.50 SIGDMACH10DONE Register (Offset = C4h) [Reset = 00000200h]

SIGDMACH10DONE is shown in Table 10-66.

Return to the Summary Table.

Table 10-66. SIGDMACH10DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH10DONE	R	200h	Signature for DMA channel 10 completion (tied to 0x0200 in HW)

10.4.51 SIGDMACH11DONE Register (Offset = C8h) [Reset = 00000400h]

SIGDMACH11DONE is shown in Table 10-67.

Return to the Summary Table.

Table 10-67. SIGDMACH11DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH11DONE	R	400h	Signature for DMA channel 11 completion (tied to 0x0040 in HW)

10.4.52 SIGDMACH12DONE Register (Offset = CCh) [Reset = 00000800h]

SIGDMACH12DONE is shown in Table 10-68.

Return to the Summary Table.

Table 10-68. SIGDMACH12DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH12DONE	R	800h	Signature for DMA channel 12 completion (tied to 0x0080 in HW)

10.4.53 SIGDMACH13DONE Register (Offset = D0h) [Reset = 00001000h]

SIGDMACH13DONE is shown in Table 10-69.

Return to the Summary Table.

Table 10-69. SIGDMACH13DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH13DONE	R	1000h	Signature for DMA channel 13 completion (tied to 0x1000 in HW)

10.4.54 SIGDMACH14DONE Register (Offset = D4h) [Reset = 00002000h]

SIGDMACH14DONE is shown in Table 10-70.

Return to the Summary Table.

Table 10-70. SIGDMACH14DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH14DONE	R	2000h	Signature for DMA channel 14 completion (tied to 0x2000 in HW)

10.4.55 SIGDMACH15DONE Register (Offset = D8h) [Reset = 00004000h]

SIGDMACH15DONE is shown in Table 10-71.

Return to the Summary Table.

Table 10-71. SIGDMACH15DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH15DONE	R	4000h	Signature for DMA channel 15 completion (tied to 0x4000 in HW)

10.4.56 SIGDMACH16DONE Register (Offset = DCh) [Reset = 00008000h]

SIGDMACH16DONE is shown in Table 10-72.

Return to the Summary Table.

Table 10-72. SIGDMACH16DONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGDMACH16DONE	R	8000h	Signature for DMA channel 16 completion (tied to 0x8000 in HW)

10.4.57 MEMACCESSERR Register (Offset = E0h) [Reset = 00000000h]

MEMACCESSERR is shown in Table 10-73.

Return to the Summary Table.

Table 10-73. MEMACCESSERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU3	R	0h	
19-16	STATERRCODE	R	0h	Reserved.TI internal
15-12	NU2	R	0h	
11-8	ERRCODEMASK	R/W	0h	Reserved.TI internal
7-4	NU1	R	0h	
3-0	ERRCODECLR		0h	Reserved.TI internal

10.4.58 FFTCLIP Register (Offset = E4h) [Reset = 00000000h]

FFTCLIP is shown in Table 10-74.

Return to the Summary Table.

Table 10-74. FFTCLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU2	R	0h	
16	CLRFFTCLIPSTAT		0h	FFTCLIPSTAT can be cleared by setting single-bit register CLRFFTCLIPSTAT, so that the saturation status indication gets cleared back to 0 and any subsequent saturation events can be freshly monitored.

Table 10-74. FFTCLIP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-10	NU1	R	0h	
9-0	FFTCLIPSTAT	R	0h	FFT Clip Status (read-only): This is a read-only status register, which indicates any saturation/clipping events that have happened in the FFT butterfly stages. Note that each of the 10 butterfly stages in the FFT can be programmed to either saturate the MSB or round the LSB. Whenever saturation of MSB is used in any stage, there is a possibility that that stage can saturate/clip samples. In that case, this saturation event is indicated in the corresponding bit in this status registert. If multiple FFTs are performed, this status register includes any saturation events happening in any of them.

10.4.59 FFTPEAKCNT Register (Offset = E8h) [Reset = 00000000h]

FFTPEAKCNT is shown in Table 10-75.

Return to the Summary Table.

Table 10-75. FFTPEAKCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU	R	0h	
11-0	FFTPEAKCNT	R	0h	CFAR Detected Peak Count: This is a read-only register that contains the number of detected peaks that are logged in the destination memory, when CFAR Engine is configured in Detected Peaks List mode.

10.4.60 HWACCREG1RD Register (Offset = ECh) [Reset = 00000000h]

HWACCREG1RD is shown in Table 10-76.

Return to the Summary Table.

Table 10-76. HWACCREG1RD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWACCREG1RD	R	0h	Reserved.TI internal

10.4.61 HWACCREG2RD Register (Offset = F0h) [Reset = 00000000h]

HWACCREG2RD is shown in Table 10-77.

Return to the Summary Table.

Table 10-77. HWACCREG2RD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWACCREG2RD	R	0h	Reserved.TI internal

10.4.62 HWACCREG3RD Register (Offset = F4h) [Reset = 00000000h]

HWACCREG3RD is shown in Table 10-78.

Return to the Summary Table.

Table 10-78. HWACCREG3RD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	HWACCREG3RD	R	0h	Reserved.TI internal

10.4.63 CMP_EGE_K0123 Register (Offset = F8h) [Reset = 00000000h]

CMP_EGE_K0123 is shown in Table 10-79.

Return to the Summary Table.

Table 10-79. CMP_EGE_K0123 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU4	R	0h	Reserved.TI internal
28-24	CMP_EGE_K3	R/W	0h	EGE K-param for the 4th accumulator
23-21	NU3	R	0h	Reserved.TI internal
20-16	CMP_EGE_K2	R/W	0h	EGE K-param for the 3rd accumulator
15-13	NU2	R	0h	Reserved.TI internal
12-8	CMP_EGE_K1	R/W	0h	EGE K-param for the 2nd accumulator
7-5	NU1	R	0h	Reserved.TI internal
4-0	CMP_EGE_K0	R/W	0h	EGE K-param for the 1st accumulator

10.4.64 CMP_EGE_K4567 Register (Offset = FCh) [Reset = 00000000h]

CMP_EGE_K4567 is shown in Table 10-80.

Return to the Summary Table.

Table 10-80. CMP_EGE_K4567 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	NU4	R	0h	Reserved.TI internal
28-24	CMP_EGE_K7	R/W	0h	EGE K-param for the 8th accumulator
23-21	NU3	R	0h	Reserved.TI internal
20-16	CMP_EGE_K6	R/W	0h	EGE K-param for the 7th accumulator
15-13	NU2	R	0h	Reserved.TI internal
12-8	CMP_EGE_K5	R/W	0h	EGE K-param for the 6th accumulator
7-5	NU1	R	0h	Reserved.TI internal
4-0	CMP_EGE_K4	R/W	0h	EGE K-param for the 5th accumulator

10.4.65 HWA_SAFETY_ENABLE Register (Offset = 100h) [Reset = 00000000h]

HWA_SAFETY_ENABLE is shown in Table 10-81.

Return to the Summary Table.

Table 10-81. HWA_SAFETY_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	NU2	R	0h	
17	FSM_LOCKSTEP_SELFTEST_EN	R/W	0h	1: Enable Selftest for Accelerator FSM
16	FSM_LOCKSTEP_EN	R/W	0h	1: Enable Lockstep for Accelerator FSM
15	OPONG_PARITY_EN	R/W	0h	1: Enable PARITY for ACCEL_MEM3
14	OPING_PARITY_EN	R/W	0h	1: Enable PARITY for ACCEL_MEM2
13	IPONG_PARITY_EN	R/W	0h	1: Enable PARITY for ACCEL_MEM1
12	IPING_PARITY_EN	R/W	0h	1: Enable PARITY for ACCEL_MEM0
11-2	NU1	R	0h	
1	PARAM_ECC_EN	R/W	0h	Not used.
0	WIN_RAM_PARITY_EN	R/W	0h	1: Enable PARITY for Window RAM

10.4.66 MEMINIT Register (Offset = 104h) [Reset = 00000000h]

MEMINIT is shown in Table 10-82.

Return to the Summary Table.

Table 10-82. MEMINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	
7	MC_ODD_INIT		0h	1: Start initialising MEM_COMPRESSION_ODD_RAM with all '0's
6	MC_EVEN_INIT		0h	1: Start initialising MEM_COMPRESSION_EVEN_RAM with all '0's
5	OPONG_INIT		0h	1: Start initialising ACCEL_MEM3 with all '0's
4	OPING_INIT		0h	1: Start initialising ACCEL_MEM2 with all '0's
3	IPONG_INIT		0h	1: Start initialising ACCEL_MEM1 with all '0's
2	IPING_INIT		0h	1: Start initialising ACCEL_MEM0 with all '0's
1	PARAM_INIT		0h	1: Start initialising Parameter set RAM with all '0's
0	WIN_RAM_INIT		0h	1: Start initialising Window RAM with all '0's

10.4.67 MEMINITDONE Register (Offset = 108h) [Reset = 00000000h]

MEMINITDONE is shown in Table 10-83.

Return to the Summary Table.

Table 10-83. MEMINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	
7	MC_ODD_INITDONE	R	0h	1: Init done status for MEM_COMPRESSION_ODD_RAM
6	MC_EVEN_INITDONE	R	0h	1: Init done status for MEM_COMPRESSION_EVEN_RAM
5	OPONG_INITDONE	R	0h	1: Init done status for ACCEL_MEM3
4	OPING_INITDONE	R	0h	1: Init done status for ACCEL_MEM2
3	IPONG_INITDONE	R	0h	1: Init done status for ACCEL_MEM1
2	IPING_INITDONE	R	0h	1: Init done status for ACCEL_MEM0
1	PARAM_INITDONE	R	0h	1: Init done status for Parameter set RAM
0	WIN_RAM_INITDONE	R	0h	1: Init done status for Window RAM

10.4.68 HWA_SAFETY_WIN_RAM_ERR_LOC Register (Offset = 10Ch) [Reset = 00000000h]

HWA_SAFETY_WIN_RAM_ERR_LOC is shown in Table 10-84.

Return to the Summary Table.

Table 10-84. HWA_SAFETY_WIN_RAM_ERR_LOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	
15-0	HWA_SAFETY_WIN_RAM_ERR_ADDR	R	0h	[Debug] Address of parity error location within Window RAM

10.4.69 HWA_SAFETY_PARAM_RAM_ERR_LOC Register (Offset = 110h) [Reset = 00000000h]

HWA_SAFETY_PARAM_RAM_ERR_LOC is shown in Table 10-85.

Return to the Summary Table.

Table 10-85. HWA_SAFETY_PARAM_RAM_ERR_LOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R	0h	Reserved.TI internal

10.4.70 HWA_SAFETY_IPING_ERR_LOC Register (Offset = 114h) [Reset = 00000000h]

HWA_SAFETY_IPING_ERR_LOC is shown in Table 10-86.

Return to the Summary Table.

Table 10-86. HWA_SAFETY_IPING_ERR_LOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	
15-0	HWA_SAFETY_IPING_ERR_ADDR	R	0h	[Debug]Address of parity error location within ACCEL_MEM0 (rows 0-1023)

10.4.71 HWA_SAFETY_IPONG_ERR_LOC Register (Offset = 118h) [Reset = 00000000h]

HWA_SAFETY_IPONG_ERR_LOC is shown in Table 10-87.

Return to the Summary Table.

Table 10-87. HWA_SAFETY_IPONG_ERR_LOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	
15-0	HWA_SAFETY_IPONG_ERR_ADDR	R	0h	[Debug]Address of parity error location within ACCEL_MEM1 (rows 0-1023)

10.4.72 HWA_SAFETY_OPING_ERR_LOC Register (Offset = 11Ch) [Reset = 00000000h]

HWA_SAFETY_OPING_ERR_LOC is shown in Table 10-88.

Return to the Summary Table.

Table 10-88. HWA_SAFETY_OPING_ERR_LOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	
15-0	HWA_SAFETY_OPING_ERR_ADDR	R	0h	[Debug]Address of parity error location within ACCEL_MEM2 (rows 0-1023)

10.4.73 HWA_SAFETY_OPONG_ERR_LOC Register (Offset = 120h) [Reset = 00000000h]

HWA_SAFETY_OPONG_ERR_LOC is shown in Table 10-89.

Return to the Summary Table.

Table 10-89. HWA_SAFETY_OPONG_ERR_LOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	
15-0	HWA_SAFETY_OPONG_ERR_ADDR	R	0h	[Debug]Address of parity error location within ACCEL_MEM3 (rows 0-1023)

10.4.74 FFTINTMEMWRDATA Register (Offset = 124h) [Reset = 00000000h]

FFTINTMEMWRDATA is shown in Table 10-90.

Return to the Summary Table.

Table 10-90. FFTINTMEMWRDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FFTINTMEMWRDATA	R/W	0h	Reserved.TI internal

10.4.75 FFTINTMEMRDDATA Register (Offset = 128h) [Reset = 00000000h]

FFTINTMEMRDDATA is shown in Table 10-91.

Return to the Summary Table.

Table 10-91. FFTINTMEMRDDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FFTINTMEMRDDATA	R	0h	Reserved.TI internal

10.4.76 HWACCREG16 Register (Offset = 12Ch) [Reset = 00000000h]

HWACCREG16 is shown in Table 10-92.

Return to the Summary Table.

Table 10-92. HWACCREG16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	NU1	R	0h	
21-17	PARAMSTOP	R/W	0h	These registers are used to control the start and stop index of the parameter-set through which the state machine loops through. The state machine starts at the parameter-set specified by PARAM_START and loads each parameter-set one after another and runs the accelerator as per that configuration. When the state machine reaches the parameter-set specified by PARAM_STOP, it loops back to the start index as specified by PARAM_START.
16-12	PARAMSTART	R/W	0h	These registers are used to control the start and stop index of the parameter-set through which the state machine loops through. The state machine starts at the parameter-set specified by PARAM_START and loads each parameter-set one after another and runs the accelerator as per that configuration. When the state machine reaches the parameter-set specified by PARAM_STOP, it loops back to the start index as specified by PARAM_START.
11-0	NLOOPS	R/W	0h	Number of loops: This register controls the number of times the State Machine will loop through the parameter-sets (from a programmed start index till a programmed end index) and run them. The maximum number of times the loop can be made is run is 4094. A value of zero programmed in this register means that the looping mechanism is disabled.

10.4.77 DCEST1I_SW Register (Offset = 130h) [Reset = 00000000h]

DCEST1I_SW is shown in Table 10-93.

Return to the Summary Table.

Table 10-93. DCEST1I_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST1I_SW	R/W	0h	This register holds the software programmed dc value I to be subtracted from incoming sample for bcnt = 0.

10.4.78 DCEST2I_SW Register (Offset = 134h) [Reset = 00000000h]

DCEST2I_SW is shown in Table 10-94.

Return to the Summary Table.

Table 10-94. DCEST2I_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST2I_SW	R/W	0h	This register holds the software programmed dc value I to be subtracted from incoming sample for bcnt = 1.

10.4.79 DCEST3I_SW Register (Offset = 138h) [Reset = 00000000h]

DCEST3I_SW is shown in Table 10-95.

Return to the Summary Table.

Table 10-95. DCEST3I_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST3I_SW	R/W	0h	This register holds the software programmed dc value I to be subtracted from incoming sample for bcnt = 2.

10.4.80 DCEST4I_SW Register (Offset = 13Ch) [Reset = 00000000h]

DCEST4I_SW is shown in Table 10-96.

Return to the Summary Table.

Table 10-96. DCEST4I_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST4I_SW	R/W	0h	This register holds the software programmed dc value I to be subtracted from incoming sample for bcnt = 3.

10.4.81 DCEST5I_SW Register (Offset = 140h) [Reset = 00000000h]

DCEST5I_SW is shown in Table 10-97.

Return to the Summary Table.

Table 10-97. DCEST5I_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST5I_SW	R/W	0h	This register holds the software programmed dc value I to be subtracted from incoming sample for bcnt = 4.

10.4.82 DCEST6I_SW Register (Offset = 144h) [Reset = 00000000h]

DCEST6I_SW is shown in Table 10-98.

Return to the Summary Table.

Table 10-98. DCEST6I_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	

Table 10-98. DCEST6I_SW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-0	DCEST6I_SW	R/W	0h	This register holds the software programmed dc value I to be subtracted from incoming sample for bcnt = 5.

10.4.83 DCEST1I Register (Offset = 148h) [Reset = 00000000h]

DCEST1I is shown in Table 10-99.

Return to the Summary Table.

Table 10-99. DCEST1I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST1I	R	0h	This register holds the estimated dc value I to be subtracted from incoming sample for bcnt =0 .

10.4.84 DCEST2I Register (Offset = 14Ch) [Reset = 00000000h]

DCEST2I is shown in Table 10-100.

Return to the Summary Table.

Table 10-100. DCEST2I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST2I	R	0h	This register holds the estimated dc value I to be subtracted from incoming sample for bcnt =1 .

10.4.85 DCEST3I Register (Offset = 150h) [Reset = 00000000h]

DCEST3I is shown in Table 10-101.

Return to the Summary Table.

Table 10-101. DCEST3I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST3I	R	0h	This register holds the estimated dc value I to be subtracted from incoming sample for bcnt =2 .

10.4.86 DCEST4I Register (Offset = 154h) [Reset = 00000000h]

DCEST4I is shown in Table 10-102.

Return to the Summary Table.

Table 10-102. DCEST4I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST4I	R	0h	This register holds the estimated dc value I to be subtracted from incoming sample for bcnt =3.

10.4.87 DCEST5I Register (Offset = 158h) [Reset = 00000000h]

DCEST5I is shown in Table 10-103.

Return to the Summary Table.

Table 10-103. DCEST5I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST5I	R	0h	This register holds the estimated dc value I to be subtracted from incoming sample for bcnt =4 .

10.4.88 DCEST6I Register (Offset = 15Ch) [Reset = 00000000h]

DCEST6I is shown in Table 10-104.

Return to the Summary Table.

Table 10-104. DCEST6I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST6I	R	0h	This register holds the estimated dc value I to be subtracted from incoming sample for bcnt =5 .

10.4.89 DC_ACC1I_LSB Register (Offset = 160h) [Reset = 00000000h]

DC_ACC1I_LSB is shown in Table 10-105.

Return to the Summary Table.

Table 10-105. DC_ACC1I_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC1I_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator I channel for bcnt=0

10.4.90 DC_ACC1I_MSB Register (Offset = 164h) [Reset = 00000000h]

DC_ACC1I_MSB is shown in Table 10-106.

Return to the Summary Table.

Table 10-106. DC_ACC1I_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC1I_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator I channel for bcnt=0

10.4.91 DC_ACC2I_LSB Register (Offset = 168h) [Reset = 00000000h]

DC_ACC2I_LSB is shown in Table 10-107.

Return to the Summary Table.

Table 10-107. DC_ACC2I_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC2I_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator I channel for bcnt=1

10.4.92 DC_ACC2I_MSB Register (Offset = 16Ch) [Reset = 00000000h]

DC_ACC2I_MSB is shown in Table 10-108.

Return to the Summary Table.

Table 10-108. DC_ACC2I_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC2I_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator I channel for bcnt=1

10.4.93 DC_ACC3I_LSB Register (Offset = 170h) [Reset = 00000000h]

DC_ACC3I_LSB is shown in Table 10-109.

Return to the Summary Table.

Table 10-109. DC_ACC3I_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC3I_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator I channel for bcnt=2

10.4.94 DC_ACC3I_MSB Register (Offset = 174h) [Reset = 00000000h]

DC_ACC3I_MSB is shown in Table 10-110.

Return to the Summary Table.

Table 10-110. DC_ACC3I_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC3I_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator I channel for bcnt=2

10.4.95 DC_ACC4I_LSB Register (Offset = 178h) [Reset = 00000000h]

DC_ACC4I_LSB is shown in Table 10-111.

Return to the Summary Table.

Table 10-111. DC_ACC4I_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC4I_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator I channel for bcnt=3

10.4.96 DC_ACC4I_MSB Register (Offset = 17Ch) [Reset = 00000000h]

DC_ACC4I_MSB is shown in Table 10-112.

Return to the Summary Table.

Table 10-112. DC_ACC4I_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC4I_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator I channel for bcnt=3

10.4.97 DC_ACC5I_LSB Register (Offset = 180h) [Reset = 00000000h]

DC_ACC5I_LSB is shown in Table 10-113.

Return to the Summary Table.

Table 10-113. DC_ACC5I_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC5I_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator I channel for bcnt=4

10.4.98 DC_ACC5I_MSB Register (Offset = 184h) [Reset = 00000000h]

DC_ACC5I_MSB is shown in Table 10-114.

Return to the Summary Table.

Table 10-114. DC_ACC5I_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC5I_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator I channel for bcnt=4

10.4.99 DC_ACC6I_LSB Register (Offset = 188h) [Reset = 00000000h]

DC_ACC6I_LSB is shown in Table 10-115.

Return to the Summary Table.

Table 10-115. DC_ACC6I_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC6I_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator I channel for bcnt=5

10.4.100 DC_ACC6I_MSB Register (Offset = 18Ch) [Reset = 00000000h]

DC_ACC6I_MSB is shown in Table 10-116.

Return to the Summary Table.

Table 10-116. DC_ACC6I_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC6I_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator I channel for bcnt=5

10.4.101 DCEST1Q_SW Register (Offset = 190h) [Reset = 00000000h]

DCEST1Q_SW is shown in Table 10-117.

Return to the Summary Table.

Table 10-117. DCEST1Q_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST1Q_SW	R/W	0h	This register holds the software programmed dc value Q to be subtracted from incoming sample for bcnt = 0.

10.4.102 DCEST2Q_SW Register (Offset = 194h) [Reset = 00000000h]

DCEST2Q_SW is shown in Table 10-118.

Return to the Summary Table.

Table 10-118. DCEST2Q_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST2Q_SW	R/W	0h	This register holds the software programmed dc value Q to be subtracted from incoming sample for bcnt = 1.

10.4.103 DCEST3Q_SW Register (Offset = 198h) [Reset = 00000000h]

DCEST3Q_SW is shown in Table 10-119.

Return to the Summary Table.

Table 10-119. DCEST3Q_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST3Q_SW	R/W	0h	This register holds the software programmed dc value Q to be subtracted from incoming sample for bcnt = 2.

10.4.104 DCEST4Q_SW Register (Offset = 19Ch) [Reset = 00000000h]

DCEST4Q_SW is shown in Table 10-120.

Return to the Summary Table.

Table 10-120. DCEST4Q_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST4Q_SW	R/W	0h	This register holds the software programmed dc value Q to be subtracted from incoming sample for bcnt = 3.

10.4.105 DCEST5Q_SW Register (Offset = 1A0h) [Reset = 00000000h]

DCEST5Q_SW is shown in Table 10-121.

Return to the Summary Table.

Table 10-121. DCEST5Q_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST5Q_SW	R/W	0h	This register holds the software programmed dc value Q to be subtracted from incoming sample for bcnt = 4.

10.4.106 DCEST6Q_SW Register (Offset = 1A4h) [Reset = 00000000h]

DCEST6Q_SW is shown in Table 10-122.

Return to the Summary Table.

Table 10-122. DCEST6Q_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	

Table 10-122. DCEST6Q_SW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-0	DCEST6Q_SW	R/W	0h	This register holds the software programmed dc value Q to be subtracted from incoming sample for bcnt = 5.

10.4.107 DCEST1Q Register (Offset = 1A8h) [Reset = 00000000h]

DCEST1Q is shown in Table 10-123.

Return to the Summary Table.

Table 10-123. DCEST1Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST1Q	R	0h	This register holds the estimated dc value Q to be subtracted from incoming sample for bcnt =0 .

10.4.108 DCEST2Q Register (Offset = 1ACh) [Reset = 00000000h]

DCEST2Q is shown in Table 10-124.

Return to the Summary Table.

Table 10-124. DCEST2Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST2Q	R	0h	This register holds the estimated dc value Q to be subtracted from incoming sample for bcnt =1 .

10.4.109 DCEST3Q Register (Offset = 1B0h) [Reset = 00000000h]

DCEST3Q is shown in Table 10-125.

Return to the Summary Table.

Table 10-125. DCEST3Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST3Q	R	0h	This register holds the estimated dc value Q to be subtracted from incoming sample for bcnt =2 .

10.4.110 DCEST4Q Register (Offset = 1B4h) [Reset = 00000000h]

DCEST4Q is shown in Table 10-126.

Return to the Summary Table.

Table 10-126. DCEST4Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST4Q	R	0h	This register holds the estimated dc value Q to be subtracted from incoming sample for bcnt =3.

10.4.111 DCEST5Q Register (Offset = 1B8h) [Reset = 00000000h]

DCEST5Q is shown in Table 10-127.

Return to the Summary Table.

Table 10-127. DCEST5Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST5Q	R	0h	This register holds the estimated dc value Q to be subtracted from incoming sample for bcnt =4 .

10.4.112 DCEST6Q Register (Offset = 1BCh) [Reset = 00000000h]

DCEST6Q is shown in Table 10-128.

Return to the Summary Table.

Table 10-128. DCEST6Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	DCEST6Q	R	0h	This register holds the estimated dc value Q to be subtracted from incoming sample for bcnt =5 .

10.4.113 DC_ACC1Q_LSB Register (Offset = 1C0h) [Reset = 00000000h]

DC_ACC1Q_LSB is shown in Table 10-129.

Return to the Summary Table.

Table 10-129. DC_ACC1Q_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC1Q_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator Q channel for bcnt=0

10.4.114 DC_ACC1Q_MSB Register (Offset = 1C4h) [Reset = 00000000h]

DC_ACC1Q_MSB is shown in Table 10-130.

Return to the Summary Table.

Table 10-130. DC_ACC1Q_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC1Q_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator Q channel for bcnt=0

10.4.115 DC_ACC2Q_LSB Register (Offset = 1C8h) [Reset = 00000000h]

DC_ACC2Q_LSB is shown in Table 10-131.

Return to the Summary Table.

Table 10-131. DC_ACC2Q_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC2Q_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator Q channel for bcnt=1

10.4.116 DC_ACC2Q_MSB Register (Offset = 1CCh) [Reset = 00000000h]

DC_ACC2Q_MSB is shown in Table 10-132.

Return to the Summary Table.

Table 10-132. DC_ACC2Q_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC2Q_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator Q channel for bcnt=1

10.4.117 DC_ACC3Q_LSB Register (Offset = 1D0h) [Reset = 00000000h]

DC_ACC3Q_LSB is shown in Table 10-133.

Return to the Summary Table.

Table 10-133. DC_ACC3Q_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC3Q_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator Q channel for bcnt=2

10.4.118 DC_ACC3Q_MSB Register (Offset = 1D4h) [Reset = 00000000h]

DC_ACC3Q_MSB is shown in Table 10-134.

Return to the Summary Table.

Table 10-134. DC_ACC3Q_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC3Q_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator Q channel for bcnt=2

10.4.119 DC_ACC4Q_LSB Register (Offset = 1D8h) [Reset = 00000000h]

DC_ACC4Q_LSB is shown in Table 10-135.

Return to the Summary Table.

Table 10-135. DC_ACC4Q_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC4Q_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator Q channel for bcnt=3

10.4.120 DC_ACC4Q_MSB Register (Offset = 1DCh) [Reset = 00000000h]

DC_ACC4Q_MSB is shown in Table 10-136.

Return to the Summary Table.

Table 10-136. DC_ACC4Q_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC4Q_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator Q channel for bcnt=3

10.4.121 DC_ACC5Q_LSB Register (Offset = 1E0h) [Reset = 00000000h]

DC_ACC5Q_LSB is shown in Table 10-137.

Return to the Summary Table.

Table 10-137. DC_ACC5Q_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC5Q_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator Q channel for bcnt=4

10.4.122 DC_ACC5Q_MSB Register (Offset = 1E4h) [Reset = 00000000h]

DC_ACC5Q_MSB is shown in Table 10-138.

Return to the Summary Table.

Table 10-138. DC_ACC5Q_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC5Q_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator Q channel for bcnt=4

10.4.123 DC_ACC6Q_LSB Register (Offset = 1E8h) [Reset = 00000000h]

DC_ACC6Q_LSB is shown in Table 10-139.

Return to the Summary Table.

Table 10-139. DC_ACC6Q_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DC_ACC6Q_LSB	R	0h	This register provides the LSB 32 bits value of DC accumulator Q channel for bcnt=5

10.4.124 DC_ACC6Q_MSB Register (Offset = 1ECh) [Reset = 00000000h]

DC_ACC6Q_MSB is shown in Table 10-140.

Return to the Summary Table.

Table 10-140. DC_ACC6Q_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DC_ACC6Q_MSB	R	0h	This register provides the MSB 4 bits value of DC accumulator Q channel for bcnt=5

10.4.125 DCACC1_CLIP Register (Offset = 1F0h) [Reset = 00000000h]

DCACC1_CLIP is shown in Table 10-141.

Return to the Summary Table.

Table 10-141. DCACC1_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCACC1_CLIP	R	0h	This register contains the clip status of both I/Q of DC accumulators for bcnt =0

10.4.126 DCACC2_CLIP Register (Offset = 1F4h) [Reset = 00000000h]

DCACC2_CLIP is shown in Table 10-142.

Return to the Summary Table.

Table 10-142. DCACC2_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCACC2_CLIP	R	0h	This register contains the clip status of both I/Q of DC accumulators for bcnt =1

10.4.127 DCACC3_CLIP Register (Offset = 1F8h) [Reset = 00000000h]

DCACC3_CLIP is shown in Table 10-143.

Return to the Summary Table.

Table 10-143. DCACC3_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCACC3_CLIP	R	0h	This register contains the clip status of both I/Q of DC accumulators for bcnt =2

10.4.128 DCACC4_CLIP Register (Offset = 1FCh) [Reset = 00000000h]

DCACC4_CLIP is shown in Table 10-144.

Return to the Summary Table.

Table 10-144. DCACC4_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCACC4_CLIP	R	0h	This register contains the clip status of both I/Q of DC accumulators for bcnt =3

10.4.129 DCACC5_CLIP Register (Offset = 200h) [Reset = 00000000h]

DCACC5_CLIP is shown in Table 10-145.

Return to the Summary Table.

Table 10-145. DCACC5_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCACC5_CLIP	R	0h	This register contains the clip status of both I/Q of DC accumulators for bcnt =4

10.4.130 DCACC6_CLIP Register (Offset = 204h) [Reset = 00000000h]

DCACC6_CLIP is shown in Table 10-146.

Return to the Summary Table.

Table 10-146. DCACC6_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	

Table 10-146. DCACC6_CLIP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	DCACC6_CLIP	R	0h	This register contains the clip status of both I/Q of DC accumulators for bcnt =5

10.4.131 DCEST1_CLIP Register (Offset = 208h) [Reset = 00000000h]

DCEST1_CLIP is shown in Table 10-147.

Return to the Summary Table.

Table 10-147. DCEST1_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCEST1_CLIP	R	0h	This register contains the clip status of both I/Q DC estimates for bcnt =0

10.4.132 DCEST2_CLIP Register (Offset = 20Ch) [Reset = 00000000h]

DCEST2_CLIP is shown in Table 10-148.

Return to the Summary Table.

Table 10-148. DCEST2_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCEST2_CLIP	R	0h	This register contains the clip status of both I/Q DC estimates for bcnt =1

10.4.133 DCEST3_CLIP Register (Offset = 210h) [Reset = 00000000h]

DCEST3_CLIP is shown in Table 10-149.

Return to the Summary Table.

Table 10-149. DCEST3_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCEST3_CLIP	R	0h	This register contains the clip status of both I/Q DC estimates for bcnt =2

10.4.134 DCEST4_CLIP Register (Offset = 214h) [Reset = 00000000h]

DCEST4_CLIP is shown in Table 10-150.

Return to the Summary Table.

Table 10-150. DCEST4_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCEST4_CLIP	R	0h	This register contains the clip status of both I/Q DC estimates for bcnt =3

10.4.135 DCEST5_CLIP Register (Offset = 218h) [Reset = 00000000h]

DCEST5_CLIP is shown in Table 10-151.

Return to the Summary Table.

Table 10-151. DCEST5_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCEST5_CLIP	R	0h	This register contains the clip status of both I/Q DC estimates for bcnt =4

10.4.136 DCEST6_CLIP Register (Offset = 21Ch) [Reset = 00000000h]

DCEST6_CLIP is shown in Table 10-152.

Return to the Summary Table.

Table 10-152. DCEST6_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCEST6_CLIP	R	0h	This register contains the clip status of both I/Q DC estimates for bcnt =5

10.4.137 DCSUB1_CLIP Register (Offset = 220h) [Reset = 00000000h]

DCSUB1_CLIP is shown in Table 10-153.

Return to the Summary Table.

Table 10-153. DCSUB1_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCSUB1_CLIP	R	0h	Indicates the DC subtraction clip status for bcnt =0

10.4.138 DCSUB2_CLIP Register (Offset = 224h) [Reset = 00000000h]

DCSUB2_CLIP is shown in Table 10-154.

Return to the Summary Table.

Table 10-154. DCSUB2_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCSUB2_CLIP	R	0h	Indicates the DC subtraction clip status for bcnt =1

10.4.139 DCSUB3_CLIP Register (Offset = 228h) [Reset = 00000000h]

DCSUB3_CLIP is shown in Table 10-155.

Return to the Summary Table.

Table 10-155. DCSUB3_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCSUB3_CLIP	R	0h	Indicates the DC subtraction clip status for bcnt =2

10.4.140 DCSUB4_CLIP Register (Offset = 22Ch) [Reset = 00000000h]

DCSUB4_CLIP is shown in Table 10-156.

Return to the Summary Table.

Table 10-156. DCSUB4_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCSUB4_CLIP	R	0h	Indicates the DC subtraction clip status for bcnt =3

10.4.141 DCSUB5_CLIP Register (Offset = 230h) [Reset = 00000000h]

DCSUB5_CLIP is shown in Table 10-157.

Return to the Summary Table.

Table 10-157. DCSUB5_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCSUB5_CLIP	R	0h	Indicates the DC subtraction clip status for bcnt =4

10.4.142 DCSUB6_CLIP Register (Offset = 234h) [Reset = 00000000h]

DCSUB6_CLIP is shown in Table 10-158.

Return to the Summary Table.

Table 10-158. DCSUB6_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCSUB6_CLIP	R	0h	Indicates the DC subtraction clip status for bcnt =5

10.4.143 DCEST_SHIFT Register (Offset = 238h) [Reset = 00000000h]

DCEST_SHIFT is shown in Table 10-159.

Return to the Summary Table.

Table 10-159. DCEST_SHIFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	DCEST_SHIFT	R/W	0h	Programmable shift applied to all 6 accumulator outputs. Cannot be bypassed. Scaled accumulator output is shifted by $2^{2+DCEST\_SHIFT}$. For DCEST_SHIFT = 15 also gives 2^{24} and not 25 (saturate at 24)

10.4.144 DCEST_SCALE Register (Offset = 23Ch) [Reset = 00000100h]

DCEST_SCALE is shown in Table 10-160.

Return to the Summary Table.

Table 10-160. DCEST_SCALE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	NU1	R	0h	

Table 10-160. DCEST_SCALE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8-0	DCEST_SCALE	R/W	100h	9-bit scale applied to all 6 accumulators. Multiplies the accumulator output by DCEST_SCALE/256.This is followed by right shift and truncation.Default value is 256 giving a scale of 1.0. Setting it to 128, gives a scale of 0.5

10.4.145 INTF_MAG_SCALE Register (Offset = 240h) [Reset = 00000008h]

INTF_MAG_SCALE is shown in Table 10-161.

Return to the Summary Table.

Table 10-161. INTF_MAG_SCALE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1	R	0h	
7-0	INTF_MAG_SCALE	R/W	8h	Unsigned scaler (5.3) applied to INTERFSUM_MAGn from interference statistics block.Default 8= scale of 1.0

10.4.146 INTF_MAG_SHIFT Register (Offset = 244h) [Reset = 00000000h]

INTF_MAG_SHIFT is shown in Table 10-162.

Return to the Summary Table.

Table 10-162. INTF_MAG_SHIFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAG_SHIFT	R/W	0h	Right shift applied after scaling – $2^{4+INTERFSUM\_MAG\_SHIFT}$. Can t be more than 2^{17}.

10.4.147 INTF_MAGDIFF_SCALE Register (Offset = 248h) [Reset = 00000008h]

INTF_MAGDIFF_SCALE is shown in Table 10-163.

Return to the Summary Table.

Table 10-163. INTF_MAGDIFF_SCALE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU1	R	0h	
7-0	INTF_MAGDIFF_SCALE	R/W	8h	Unsigned scaler (5.3) applied to INTERFSUM_MAGDIFFn from interference statistics block.Default 8= scale of 1.0

10.4.148 INTF_MAGDIFF_SHIFT Register (Offset = 24Ch) [Reset = 00000000h]

INTF_MAGDIFF_SHIFT is shown in Table 10-164.

Return to the Summary Table.

Table 10-164. INTF_MAGDIFF_SHIFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGDIFF_SHIFT	R/W	0h	Right shift applied after scaling – $2^{4+INTERFSUM\_MAGDIFF\_SHIFT}$. Can t be more than 2^{17}.

10.4.149 INTF_FRAME_ZEROCOUNT Register (Offset = 250h) [Reset = 00000000h]

INTF_FRAME_ZEROCOUNT is shown in Table 10-165.

Return to the Summary Table.

Table 10-165. INTF_FRAME_ZEROCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	NU1	R	0h	
19-0	INTF_FRAME_ZEROCOUNT	R	0h	Number of samples that exceeded the threshold in a frame

10.4.150 INTF_CHIRP_ZEROCOUNT Register (Offset = 254h) [Reset = 00000000h]

INTF_CHIRP_ZEROCOUNT is shown in Table 10-166.

Return to the Summary Table.

Table 10-166. INTF_CHIRP_ZEROCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU1	R	0h	
11-0	INTF_CHIRP_ZEROCOUNT	R	0h	Number of samples that exceeded the threshold in a chirp

10.4.151 INTF_MAGTHRESH1_SW Register (Offset = 258h) [Reset = 00000000h]

INTF_MAGTHRESH1_SW is shown in Table 10-167.

Return to the Summary Table.

Table 10-167. INTF_MAGTHRESH1_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH1_SW	R/W	0h	This register provides software programmed interference magnitude threshold value for bcnt =0

10.4.152 INTF_MAGTHRESH2_SW Register (Offset = 25Ch) [Reset = 00000000h]

INTF_MAGTHRESH2_SW is shown in Table 10-168.

Return to the Summary Table.

Table 10-168. INTF_MAGTHRESH2_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH2_SW	R/W	0h	This register provides software programmed interference magnitude threshold value for bcnt =1

10.4.153 INTF_MAGTHRESH3_SW Register (Offset = 260h) [Reset = 00000000h]

INTF_MAGTHRESH3_SW is shown in Table 10-169.

Return to the Summary Table.

Table 10-169. INTF_MAGTHRESH3_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	

Table 10-169. INTF_MAGTHRESH3_SW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-0	INTF_MAGTHRESH3_SW	R/W	0h	This register provides software programmed interference magnitude threshold value for bcnt =2

10.4.154 INTF_MAGTHRESH4_SW Register (Offset = 264h) [Reset = 00000000h]

INTF_MAGTHRESH4_SW is shown in Table 10-170.

Return to the Summary Table.

Table 10-170. INTF_MAGTHRESH4_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH4_SW	R/W	0h	This register provides software programmed interference magnitude threshold value for bcnt =3

10.4.155 INTF_MAGTHRESH5_SW Register (Offset = 268h) [Reset = 00000000h]

INTF_MAGTHRESH5_SW is shown in Table 10-171.

Return to the Summary Table.

Table 10-171. INTF_MAGTHRESH5_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH5_SW	R/W	0h	This register provides software programmed interference magnitude threshold value for bcnt =4

10.4.156 INTF_MAGTHRESH6_SW Register (Offset = 26Ch) [Reset = 00000000h]

INTF_MAGTHRESH6_SW is shown in Table 10-172.

Return to the Summary Table.

Table 10-172. INTF_MAGTHRESH6_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH6_SW	R/W	0h	This register provides software programmed interference magnitude threshold value for bcnt =5

10.4.157 INTF_MAGDIFFTHRESH1_SW Register (Offset = 270h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH1_SW is shown in Table 10-173.

Return to the Summary Table.

Table 10-173. INTF_MAGDIFFTHRESH1_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH1_SW	R/W	0h	This register provides software programmed interference magnitude difference threshold value for bcnt =0

10.4.158 INTF_MAGDIFFTHRESH2_SW Register (Offset = 274h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH2_SW is shown in Table 10-174.

Return to the Summary Table.

Table 10-174. INTF_MAGDIFFTHRESH2_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH2_SW	R/W	0h	This register provides software programmed interference magnitude difference threshold value for bcnt =1

10.4.159 INTF_MAGDIFFTHRESH3_SW Register (Offset = 278h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH3_SW is shown in Table 10-175.

Return to the Summary Table.

Table 10-175. INTF_MAGDIFFTHRESH3_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH3_SW	R/W	0h	This register provides software programmed interference magnitude difference threshold value for bcnt =2

10.4.160 INTF_MAGDIFFTHRESH4_SW Register (Offset = 27Ch) [Reset = 00000000h]

INTF_MAGDIFFTHRESH4_SW is shown in Table 10-176.

Return to the Summary Table.

Table 10-176. INTF_MAGDIFFTHRESH4_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH4_SW	R/W	0h	This register provides software programmed interference magnitude difference threshold value for bcnt =3

10.4.161 INTF_MAGDIFFTHRESH5_SW Register (Offset = 280h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH5_SW is shown in Table 10-177.

Return to the Summary Table.

Table 10-177. INTF_MAGDIFFTHRESH5_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH5_SW	R/W	0h	This register provides software programmed interference magnitude difference threshold value for bcnt =4

10.4.162 INTF_MAGDIFFTHRESH6_SW Register (Offset = 284h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH6_SW is shown in Table 10-178.

Return to the Summary Table.

Table 10-178. INTF_MAGDIFFTHRESH6_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH6_SW	R/W	0h	This register provides software programmed interference magnitude difference threshold value for bcnt =5

10.4.163 INTF_MAGACC1_LSB Register (Offset = 288h) [Reset = 00000000h]

INTF_MAGACC1_LSB is shown in Table 10-179.

Return to the Summary Table.

Table 10-179. INTF_MAGACC1_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGACC1_LSB	R	0h	This register contains the accumulator value of the interference magnitude (LSB 32 bits) for bcnt = 0

10.4.164 INTF_MAGACC1_MSB Register (Offset = 28Ch) [Reset = 00000000h]

INTF_MAGACC1_MSB is shown in Table 10-180.

Return to the Summary Table.

Table 10-180. INTF_MAGACC1_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGACC1_MSB	R	0h	This register contains the accumulator value of the interference magnitude (MSB 4 bits) for bcnt = 0

10.4.165 INTF_MAGACC2_LSB Register (Offset = 290h) [Reset = 00000000h]

INTF_MAGACC2_LSB is shown in Table 10-181.

Return to the Summary Table.

Table 10-181. INTF_MAGACC2_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGACC2_LSB	R	0h	This register contains the accumulator value of the interference magnitude (LSB 32 bits) for bcnt = 1

10.4.166 INTF_MAGACC2_MSB Register (Offset = 294h) [Reset = 00000000h]

INTF_MAGACC2_MSB is shown in Table 10-182.

Return to the Summary Table.

Table 10-182. INTF_MAGACC2_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGACC2_MSB	R	0h	This register contains the accumulator value of the interference magnitude (MSB 4 bits) for bcnt = 1

10.4.167 INTF_MAGACC3_LSB Register (Offset = 298h) [Reset = 00000000h]

INTF_MAGACC3_LSB is shown in Table 10-183.

Return to the Summary Table.

Table 10-183. INTF_MAGACC3_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGACC3_LSB	R	0h	This register contains the accumulator value of the interference magnitude (LSB 32 bits) for bcnt = 2

10.4.168 INTF_MAGACC3_MSB Register (Offset = 29Ch) [Reset = 00000000h]

INTF_MAGACC3_MSB is shown in Table 10-184.

Return to the Summary Table.

Table 10-184. INTF_MAGACC3_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGACC3_MSB	R	0h	This register contains the accumulator value of the interference magnitude (MSB 4 bits) for bcnt = 2

10.4.169 INTF_MAGACC4_LSB Register (Offset = 2A0h) [Reset = 00000000h]

INTF_MAGACC4_LSB is shown in Table 10-185.

Return to the Summary Table.

Table 10-185. INTF_MAGACC4_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGACC4_LSB	R	0h	This register contains the accumulator value of the interference magnitude (LSB 32 bits) for bcnt = 3

10.4.170 INTF_MAGACC4_MSB Register (Offset = 2A4h) [Reset = 00000000h]

INTF_MAGACC4_MSB is shown in Table 10-186.

Return to the Summary Table.

Table 10-186. INTF_MAGACC4_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGACC4_MSB	R	0h	This register contains the accumulator value of the interference magnitude (MSB 4 bits) for bcnt = 3

10.4.171 INTF_MAGACC5_LSB Register (Offset = 2A8h) [Reset = 00000000h]

INTF_MAGACC5_LSB is shown in Table 10-187.

Return to the Summary Table.

Table 10-187. INTF_MAGACC5_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGACC5_LSB	R	0h	This register contains the accumulator value of the interference magnitude (LSB 32 bits) for bcnt = 4

10.4.172 INTF_MAGACC5_MSB Register (Offset = 2ACh) [Reset = 00000000h]

INTF_MAGACC5_MSB is shown in Table 10-188.

Return to the Summary Table.

Table 10-188. INTF_MAGACC5_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGACC5_MSB	R	0h	This register contains the accumulator value of the interference magnitude (MSB 4 bits) for bcnt = 4

10.4.173 INTF_MAGACC6_LSB Register (Offset = 2B0h) [Reset = 00000000h]

INTF_MAGACC6_LSB is shown in Table 10-189.

Return to the Summary Table.

Table 10-189. INTF_MAGACC6_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGACC6_LSB	R	0h	This register contains the accumulator value of the interference magnitude (LSB 32 bits) for bcnt = 5

10.4.174 INTF_MAGACC6_MSB Register (Offset = 2B4h) [Reset = 00000000h]

INTF_MAGACC6_MSB is shown in Table 10-190.

Return to the Summary Table.

Table 10-190. INTF_MAGACC6_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGACC6_MSB	R	0h	This register contains the accumulator value of the interference magnitude (MSB 4 bits) for bcnt = 5

10.4.175 INTF_MAGDIFFACC1_LSB Register (Offset = 2B8h) [Reset = 00000000h]

INTF_MAGDIFFACC1_LSB is shown in Table 10-191.

Return to the Summary Table.

Table 10-191. INTF_MAGDIFFACC1_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGDIFFACC1_LSB	R	0h	This register contains the accumulator value of the interference magnitude difference (LSB 32 bits) for bcnt = 0

10.4.176 INTF_MAGDIFFACC1_MSB Register (Offset = 2BCh) [Reset = 00000000h]

INTF_MAGDIFFACC1_MSB is shown in Table 10-192.

Return to the Summary Table.

Table 10-192. INTF_MAGDIFFACC1_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGDIFFACC1_MSB	R	0h	This register contains the accumulator value of the interference magnitude difference (MSB 4 bits) for bcnt = 0

10.4.177 INTF_MAGDIFFACC2_LSB Register (Offset = 2C0h) [Reset = 00000000h]

INTF_MAGDIFFACC2_LSB is shown in Table 10-193.

Return to the Summary Table.

Table 10-193. INTF_MAGDIFFACC2_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGDIFFACC2_LSB	R	0h	This register contains the accumulator value of the interference magnitude difference (LSB 32 bits) for bcnt = 1

10.4.178 INTF_MAGDIFFACC2_MSB Register (Offset = 2C4h) [Reset = 00000000h]

INTF_MAGDIFFACC2_MSB is shown in Table 10-194.

Return to the Summary Table.

Table 10-194. INTF_MAGDIFFACC2_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGDIFFACC2_MSB	R	0h	This register contains the accumulator value of the interference magnitude difference (MSB 4 bits) for bcnt = 1

10.4.179 INTF_MAGDIFFACC3_LSB Register (Offset = 2C8h) [Reset = 00000000h]

INTF_MAGDIFFACC3_LSB is shown in Table 10-195.

Return to the Summary Table.

Table 10-195. INTF_MAGDIFFACC3_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGDIFFACC3_LSB	R	0h	This register contains the accumulator value of the interference magnitude difference (LSB 32 bits) for bcnt = 2

10.4.180 INTF_MAGDIFFACC3_MSB Register (Offset = 2CCh) [Reset = 00000000h]

INTF_MAGDIFFACC3_MSB is shown in Table 10-196.

Return to the Summary Table.

Table 10-196. INTF_MAGDIFFACC3_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGDIFFACC3_MSB	R	0h	This register contains the accumulator value of the interference magnitude difference (MSB 4 bits) for bcnt = 2

10.4.181 INTF_MAGDIFFACC4_LSB Register (Offset = 2D0h) [Reset = 00000000h]

INTF_MAGDIFFACC4_LSB is shown in Table 10-197.

Return to the Summary Table.

Table 10-197. INTF_MAGDIFFACC4_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGDIFFACC4_LSB	R	0h	This register contains the accumulator value of the interference magnitude difference (LSB 32 bits) for bcnt = 3

10.4.182 INTF_MAGDIFFACC4_MSB Register (Offset = 2D4h) [Reset = 00000000h]

INTF_MAGDIFFACC4_MSB is shown in Table 10-198.

Return to the Summary Table.

Table 10-198. INTF_MAGDIFFACC4_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGDIFFACC4_MSB	R	0h	This register contains the accumulator value of the interference magnitude difference (MSB 4 bits) for bcnt = 3

10.4.183 INTF_MAGDIFFACC5_LSB Register (Offset = 2D8h) [Reset = 00000000h]

INTF_MAGDIFFACC5_LSB is shown in Table 10-199.

Return to the Summary Table.

Table 10-199. INTF_MAGDIFFACC5_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGDIFFACC5_LSB	R	0h	This register contains the accumulator value of the interference magnitude difference (LSB 32 bits) for bcnt = 4

10.4.184 INTF_MAGDIFFACC5_MSB Register (Offset = 2DCh) [Reset = 00000000h]

INTF_MAGDIFFACC5_MSB is shown in Table 10-200.

Return to the Summary Table.

Table 10-200. INTF_MAGDIFFACC5_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGDIFFACC5_MSB	R	0h	This register contains the accumulator value of the interference magnitude difference (MSB 4 bits) for bcnt = 4

10.4.185 INTF_MAGDIFFACC6_LSB Register (Offset = 2E0h) [Reset = 00000000h]

INTF_MAGDIFFACC6_LSB is shown in Table 10-201.

Return to the Summary Table.

Table 10-201. INTF_MAGDIFFACC6_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INTF_MAGDIFFACC6_LSB	R	0h	This register contains the accumulator value of the interference magnitude difference (LSB 32 bits) for bcnt = 5

10.4.186 INTF_MAGDIFFACC6_MSB Register (Offset = 2E4h) [Reset = 00000000h]

INTF_MAGDIFFACC6_MSB is shown in Table 10-202.

Return to the Summary Table.

Table 10-202. INTF_MAGDIFFACC6_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	NU1	R	0h	
3-0	INTF_MAGDIFFACC6_MSB	R	0h	This register contains the accumulator value of the interference magnitude difference (MSB 4 bits) for bcnt = 5

10.4.187 INTF_MAGACC1_CLIP Register (Offset = 2E8h) [Reset = 00000000h]

INTF_MAGACC1_CLIP is shown in Table 10-203.

Return to the Summary Table.

Table 10-203. INTF_MAGACC1_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGACC1_CLIP	R	0h	Interference magnitude accumulator clip status

10.4.188 INTF_MAGACC2_CLIP Register (Offset = 2ECh) [Reset = 00000000h]

INTF_MAGACC2_CLIP is shown in Table 10-204.

Return to the Summary Table.

Table 10-204. INTF_MAGACC2_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGACC2_CLIP	R	0h	Interference magnitude accumulator clip status

10.4.189 INTF_MAGACC3_CLIP Register (Offset = 2F0h) [Reset = 00000000h]

INTF_MAGACC3_CLIP is shown in Table 10-205.

Return to the Summary Table.

Table 10-205. INTF_MAGACC3_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGACC3_CLIP	R	0h	Interference magnitude accumulator clip status

10.4.190 INTF_MAGACC4_CLIP Register (Offset = 2F4h) [Reset = 00000000h]

INTF_MAGACC4_CLIP is shown in Table 10-206.

Return to the Summary Table.

Table 10-206. INTF_MAGACC4_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGACC4_CLIP	R	0h	Interference magnitude accumulator clip status

10.4.191 INTF_MAGACC5_CLIP Register (Offset = 2F8h) [Reset = 00000000h]

INTF_MAGACC5_CLIP is shown in Table 10-207.

Return to the Summary Table.

Table 10-207. INTF_MAGACC5_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGACC5_CLIP	R	0h	Interference magnitude accumulator clip status

10.4.192 INTF_MAGACC6_CLIP Register (Offset = 2FCh) [Reset = 00000000h]

INTF_MAGACC6_CLIP is shown in Table 10-208.

Return to the Summary Table.

Table 10-208. INTF_MAGACC6_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGACC6_CLIP	R	0h	Interference magnitude accumulator clip status

10.4.193 INTF_MAGDIFFACC1_CLIP Register (Offset = 300h) [Reset = 00000000h]

INTF_MAGDIFFACC1_CLIP is shown in Table 10-209.

Return to the Summary Table.

Table 10-209. INTF_MAGDIFFACC1_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFACC1_CLIP	R	0h	Interference magnitude difference accumulator clip status

10.4.194 INTF_MAGDIFFACC2_CLIP Register (Offset = 304h) [Reset = 00000000h]

INTF_MAGDIFFACC2_CLIP is shown in Table 10-210.

Return to the Summary Table.

Table 10-210. INTF_MAGDIFFACC2_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFACC2_CLIP	R	0h	Interference magnitude difference accumulator clip status

10.4.195 INTF_MAGDIFFACC3_CLIP Register (Offset = 308h) [Reset = 00000000h]

INTF_MAGDIFFACC3_CLIP is shown in Table 10-211.

Return to the Summary Table.

Table 10-211. INTF_MAGDIFFACC3_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFACC3_CLIP	R	0h	Interference magnitude difference accumulator clip status

10.4.196 INTF_MAGDIFFACC4_CLIP Register (Offset = 30Ch) [Reset = 00000000h]

INTF_MAGDIFFACC4_CLIP is shown in Table 10-212.

Return to the Summary Table.

Table 10-212. INTF_MAGDIFFACC4_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFACC4_CLIP	R	0h	Interference magnitude difference accumulator clip status

10.4.197 INTF_MAGDIFFACC5_CLIP Register (Offset = 310h) [Reset = 00000000h]

INTF_MAGDIFFACC5_CLIP is shown in Table 10-213.

Return to the Summary Table.

Table 10-213. INTF_MAGDIFFACC5_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	

Table 10-213. INTF_MAGDIFFACC5_CLIP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	INTF_MAGDIFFACC5_CLIP	R	0h	Interference magnitude difference accumulator clip status

10.4.198 INTF_MAGDIFFACC6_CLIP Register (Offset = 314h) [Reset = 00000000h]

INTF_MAGDIFFACC6_CLIP is shown in Table 10-214.

Return to the Summary Table.

Table 10-214. INTF_MAGDIFFACC6_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFACC6_CLIP	R	0h	Interference magnitude difference accumulator clip status

10.4.199 INTF_MAGTHRESH1 Register (Offset = 318h) [Reset = 00000000h]

INTF_MAGTHRESH1 is shown in Table 10-215.

Return to the Summary Table.

Table 10-215. INTF_MAGTHRESH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH1	R	0h	Indicates interference magnitude threshold by interference statistics for bcnt =0

10.4.200 INTF_MAGTHRESH2 Register (Offset = 31Ch) [Reset = 00000000h]

INTF_MAGTHRESH2 is shown in Table 10-216.

Return to the Summary Table.

Table 10-216. INTF_MAGTHRESH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH2	R	0h	Indicates interference magnitude threshold by interference statistics for bcnt =1

10.4.201 INTF_MAGTHRESH3 Register (Offset = 320h) [Reset = 00000000h]

INTF_MAGTHRESH3 is shown in Table 10-217.

Return to the Summary Table.

Table 10-217. INTF_MAGTHRESH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH3	R	0h	Indicates interference magnitude threshold by interference statistics for bcnt =2

10.4.202 INTF_MAGTHRESH4 Register (Offset = 324h) [Reset = 00000000h]

INTF_MAGTHRESH4 is shown in Table 10-218.

Return to the Summary Table.

Table 10-218. INTF_MAGTHRESH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH4	R	0h	Indicates interference magnitude threshold by interference statistics for bcnt =3

10.4.203 INTF_MAGTHRESH5 Register (Offset = 328h) [Reset = 00000000h]

INTF_MAGTHRESH5 is shown in Table 10-219.

Return to the Summary Table.

Table 10-219. INTF_MAGTHRESH5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH5	R	0h	Indicates interference magnitude threshold by interference statistics for bcnt =4

10.4.204 INTF_MAGTHRESH6 Register (Offset = 32Ch) [Reset = 00000000h]

INTF_MAGTHRESH6 is shown in Table 10-220.

Return to the Summary Table.

Table 10-220. INTF_MAGTHRESH6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGTHRESH6	R	0h	Indicates interference magnitude threshold by interference statistics for bcnt =5

10.4.205 INTF_MAGDIFFTHRESH1 Register (Offset = 330h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH1 is shown in Table 10-221.

Return to the Summary Table.

Table 10-221. INTF_MAGDIFFTHRESH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH1	R	0h	Indicates interference magnitude difference threshold by interference statistics for bcnt =0

10.4.206 INTF_MAGDIFFTHRESH2 Register (Offset = 334h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH2 is shown in Table 10-222.

Return to the Summary Table.

Table 10-222. INTF_MAGDIFFTHRESH2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH2	R	0h	Indicates interference magnitude difference threshold by interference statistics for bcnt =1

10.4.207 INTF_MAGDIFFTHRESH3 Register (Offset = 338h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH3 is shown in Table 10-223.

Return to the Summary Table.

Table 10-223. INTF_MAGDIFFTHRESH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH3	R	0h	Indicates interference magnitude difference threshold by interference statistics for bcnt =2

10.4.208 INTF_MAGDIFFTHRESH4 Register (Offset = 33Ch) [Reset = 00000000h]

INTF_MAGDIFFTHRESH4 is shown in Table 10-224.

Return to the Summary Table.

Table 10-224. INTF_MAGDIFFTHRESH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH4	R	0h	Indicates interference magnitude difference threshold by interference statistics for bcnt =3

10.4.209 INTF_MAGDIFFTHRESH5 Register (Offset = 340h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH5 is shown in Table 10-225.

Return to the Summary Table.

Table 10-225. INTF_MAGDIFFTHRESH5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH5	R	0h	Indicates interference magnitude difference threshold by interference statistics for bcnt =4

10.4.210 INTF_MAGDIFFTHRESH6 Register (Offset = 344h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH6 is shown in Table 10-226.

Return to the Summary Table.

Table 10-226. INTF_MAGDIFFTHRESH6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_MAGDIFFTHRESH6	R	0h	Indicates interference magnitude difference threshold by interference statistics for bcnt =5

10.4.211 INTF_SUMMAGTHRESH Register (Offset = 348h) [Reset = 00000000h]

INTF_SUMMAGTHRESH is shown in Table 10-227.

Return to the Summary Table.

Table 10-227. INTF_SUMMAGTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	

Table 10-227. INTF_SUMMAGTHRESH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-0	INTF_SUMMAGTHRESH	R	0h	Indicates the sum of mag values only Configured BCNT mag values are added

10.4.212 INTF_SUMMAGDIFFTHRESH Register (Offset = 34Ch) [Reset = 00000000h]

INTF_SUMMAGDIFFTHRESH is shown in Table 10-228.

Return to the Summary Table.

Table 10-228. INTF_SUMMAGDIFFTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NU1	R	0h	
23-0	INTF_SUMMAGDIFFTHRESH	R	0h	Indicates the sum of magdiff values only Configured BCNT magdiff values are added

10.4.213 INTF_SUMMAGTHRESH_CLIP Register (Offset = 350h) [Reset = 00000000h]

INTF_SUMMAGTHRESH_CLIP is shown in Table 10-229.

Return to the Summary Table.

Table 10-229. INTF_SUMMAGTHRESH_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_SUMMAGTHRESH_CLIP	R	0h	Indicates the clip status of sum of magnitude threshold values

10.4.214 INTF_SUMMAGDIFFTHRESH_CLIP Register (Offset = 354h) [Reset = 00000000h]

INTF_SUMMAGDIFFTHRESH_CLIP is shown in Table 10-230.

Return to the Summary Table.

Table 10-230. INTF_SUMMAGDIFFTHRESH_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_SUMMAGDIFFTHRESH_CLIP	R	0h	Indicates the clip status of sum of magnitude difference threshold values

10.4.215 CMULTSCALE1I Register (Offset = 358h) [Reset = 00000000h]

CMULTSCALE1I is shown in Table 10-231.

Return to the Summary Table.

Table 10-231. CMULTSCALE1I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU	R	0h	
20-0	CMULTSCALE1I	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.216 CMULTSCALE2I Register (Offset = 35Ch) [Reset = 00000000h]

CMULTSCALE2I is shown in Table 10-232.

Return to the Summary Table.

Table 10-232. CMULTSCALE2I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE2I	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.217 CMULTSCALE3I Register (Offset = 360h) [Reset = 00000000h]

CMULTSCALE3I is shown in Table 10-233.

Return to the Summary Table.

Table 10-233. CMULTSCALE3I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE3I	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.218 CMULTSCALE4I Register (Offset = 364h) [Reset = 00000000h]

CMULTSCALE4I is shown in Table 10-234.

Return to the Summary Table.

Table 10-234. CMULTSCALE4I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE4I	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.219 CMULTSCALE5I Register (Offset = 368h) [Reset = 00000000h]

CMULTSCALE5I is shown in Table 10-235.

Return to the Summary Table.

Table 10-235. CMULTSCALE5I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	

Table 10-235. CMULTSCALE5I Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20-0	CMULTSCALE5I	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.220 CMULTSCALE6I Register (Offset = 36Ch) [Reset = 00000000h]

CMULTSCALE6I is shown in Table 10-236.

Return to the Summary Table.

Table 10-236. CMULTSCALE6I Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE6I	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.221 CMULTSCALE1Q Register (Offset = 370h) [Reset = 00000000h]

CMULTSCALE1Q is shown in Table 10-237.

Return to the Summary Table.

Table 10-237. CMULTSCALE1Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU	R	0h	
20-0	CMULTSCALE1Q	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.222 CMULTSCALE2Q Register (Offset = 374h) [Reset = 00000000h]

CMULTSCALE2Q is shown in Table 10-238.

Return to the Summary Table.

Table 10-238. CMULTSCALE2Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE2Q	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.223 CMULTSCALE3Q Register (Offset = 378h) [Reset = 00000000h]

CMULTSCALE3Q is shown in Table 10-239.

Return to the Summary Table.

Table 10-239. CMULTSCALE3Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE3Q	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.224 CMULTSCALE4Q Register (Offset = 37Ch) [Reset = 00000000h]

CMULTSCALE4Q is shown in Table 10-240.

Return to the Summary Table.

Table 10-240. CMULTSCALE4Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE4Q	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.225 CMULTSCALE5Q Register (Offset = 380h) [Reset = 00000000h]

CMULTSCALE5Q is shown in Table 10-241.

Return to the Summary Table.

Table 10-241. CMULTSCALE5Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	
20-0	CMULTSCALE5Q	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.226 CMULTSCALE6Q Register (Offset = 384h) [Reset = 00000000h]

CMULTSCALE6Q is shown in Table 10-242.

Return to the Summary Table.

Table 10-242. CMULTSCALE6Q Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	NU1	R	0h	

Table 10-242. CMULTSCALE6Q Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20-0	CMULTSCALE6Q	R/W	0h	In CMULT_MODE : 101 , the input samples are multiplied by a different complex scalar CMULTSCALE1I, CMULTSCALE1Q to CMULTSCALE6I, CMULTSCALE6Q per-iteration based on REG_BCNT. Else, a constant complex scalar CMULTSCALE1I and CMULTSCALEQI is applied to all sample across all iterations.

10.4.227 CLR_MISC_CLIP Register (Offset = 388h) [Reset = 00000000h]

CLR_MISC_CLIP is shown in Table 10-243.

Return to the Summary Table.

Table 10-243. CLR_MISC_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	CLR_MISC_CLIP		0h	This clears the following clip register :- dc_acc_clip_status dc_est_clip_status intf_stats_mag_accumulator_clip_status Intf_stats_magdiff_accumulator_clip_status intf_stats_thresh_mag_clip_status intf_stats_thresh_magdiff_clip_status ip_formatter_clip_status op_formatter_clip_status intf_stats_sum_mag_val_clip_status intf_stats_sum_magdiff_val_clip_status Its a self clearing bit

10.4.228 FFTINTMEMADDR Register (Offset = 38Ch) [Reset = 00000000h]

FFTINTMEMADDR is shown in Table 10-244.

Return to the Summary Table.

Table 10-244. FFTINTMEMADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NU3	R	0h	Reserved.TI internal
24	FFT_INT_MEM_RD	R/W	0h	Reserved.TI internal
23-17	NU2	R	0h	Reserved.TI internal
16	FFT_INT_MEM_EN		0h	Reserved.TI internal
15-12	NU1	R	0h	Reserved.TI internal
11-9	FFT_INT_MEM_SEL	R/W	0h	Reserved.TI internal
8-0	FFT_INT_MEM_ADDR	R/W	0h	Reserved.TI internal

10.4.229 INTF_STATS_RESET_SW Register (Offset = 390h) [Reset = 00000000h]

INTF_STATS_RESET_SW is shown in Table 10-245.

Return to the Summary Table.

Table 10-245. INTF_STATS_RESET_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_STATS_RESET_SW		0h	SW reset for Interference statistics module. Its a self clearing bit.

10.4.230 DCEST_RESET_SW Register (Offset = 394h) [Reset = 00000000h]

DCEST_RESET_SW is shown in Table 10-246.

Return to the Summary Table.

Table 10-246. DCEST_RESET_SW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	DCEST_RESET_SW		0h	Reset for all 6 DC estimation accumulators.Its a self clearing bit.

10.4.231 IP_OP_FORMATTER_CLIP_STATUS Register (Offset = 398h) [Reset = 00000000h]

IP_OP_FORMATTER_CLIP_STATUS is shown in Table 10-247.

Return to the Summary Table.

Table 10-247. IP_OP_FORMATTER_CLIP_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	NU2	R	0h	
16	OP_FORMATTER_CLIP_STATUS	R	0h	Indicates output formatter clip status
15-1	NU1	R	0h	
0	IP_FORMATTER_CLIP_STATUS	R	0h	Indicates input formatter clip status

10.4.232 INTF_MAGTHRESH1_CLIP Register (Offset = 39Ch) [Reset = 00000000h]

INTF_MAGTHRESH1_CLIP is shown in Table 10-248.

Return to the Summary Table.

Table 10-248. INTF_MAGTHRESH1_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGTHRESH1_CLIP	R	0h	Interference magnitude threshold clip status

10.4.233 INTF_MAGTHRESH2_CLIP Register (Offset = 3A0h) [Reset = 00000000h]

INTF_MAGTHRESH2_CLIP is shown in Table 10-249.

Return to the Summary Table.

Table 10-249. INTF_MAGTHRESH2_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGTHRESH2_CLIP	R	0h	Interference magnitude threshold clip status

10.4.234 INTF_MAGTHRESH3_CLIP Register (Offset = 3A4h) [Reset = 00000000h]

INTF_MAGTHRESH3_CLIP is shown in Table 10-250.

Return to the Summary Table.

Table 10-250. INTF_MAGTHRESH3_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	

Table 10-250. INTF_MAGTHRESH3_CLIP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	INTF_MAGTHRESH3_CLIP	R	0h	Interference magnitude threshold clip status

10.4.235 INTF_MAGTHRESH4_CLIP Register (Offset = 3A8h) [Reset = 00000000h]

INTF_MAGTHRESH4_CLIP is shown in Table 10-251.

Return to the Summary Table.

Table 10-251. INTF_MAGTHRESH4_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGTHRESH4_CLIP	R	0h	Interference magnitude threshold clip status

10.4.236 INTF_MAGTHRESH5_CLIP Register (Offset = 3ACh) [Reset = 00000000h]

INTF_MAGTHRESH5_CLIP is shown in Table 10-252.

Return to the Summary Table.

Table 10-252. INTF_MAGTHRESH5_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGTHRESH5_CLIP	R	0h	Interference magnitude threshold clip status

10.4.237 INTF_MAGTHRESH6_CLIP Register (Offset = 3B0h) [Reset = 00000000h]

INTF_MAGTHRESH6_CLIP is shown in Table 10-253.

Return to the Summary Table.

Table 10-253. INTF_MAGTHRESH6_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGTHRESH6_CLIP	R	0h	Interference magnitude threshold clip status

10.4.238 INTF_MAGDIFFTHRESH1_CLIP Register (Offset = 3B4h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH1_CLIP is shown in Table 10-254.

Return to the Summary Table.

Table 10-254. INTF_MAGDIFFTHRESH1_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFTHRESH1_CLIP	R	0h	Interference magnitude difference threshold clip status

10.4.239 INTF_MAGDIFFTHRESH2_CLIP Register (Offset = 3B8h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH2_CLIP is shown in Table 10-255.

Return to the Summary Table.

Table 10-255. INTF_MAGDIFFTHRESH2_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFTHRESH2_CLIP	R	0h	Interference magnitude difference threshold clip status

10.4.240 INTF_MAGDIFFTHRESH3_CLIP Register (Offset = 3BCh) [Reset = 00000000h]

INTF_MAGDIFFTHRESH3_CLIP is shown in Table 10-256.

Return to the Summary Table.

Table 10-256. INTF_MAGDIFFTHRESH3_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFTHRESH3_CLIP	R	0h	Interference magnitude difference threshold clip status

10.4.241 INTF_MAGDIFFTHRESH4_CLIP Register (Offset = 3C0h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH4_CLIP is shown in Table 10-257.

Return to the Summary Table.

Table 10-257. INTF_MAGDIFFTHRESH4_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFTHRESH4_CLIP	R	0h	Interference magnitude difference threshold clip status

10.4.242 INTF_MAGDIFFTHRESH5_CLIP Register (Offset = 3C4h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH5_CLIP is shown in Table 10-258.

Return to the Summary Table.

Table 10-258. INTF_MAGDIFFTHRESH5_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFTHRESH5_CLIP	R	0h	Interference magnitude difference threshold clip status

10.4.243 INTF_MAGDIFFTHRESH6_CLIP Register (Offset = 3C8h) [Reset = 00000000h]

INTF_MAGDIFFTHRESH6_CLIP is shown in Table 10-259.

Return to the Summary Table.

Table 10-259. INTF_MAGDIFFTHRESH6_CLIP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU1	R	0h	
0	INTF_MAGDIFFTHRESH6_CLIP	R	0h	Interference magnitude difference threshold clip status

10.4.244 HWA_SAFETY_ERR_MASK Register (Offset = 3CCh) [Reset = 00000000h]

HWA_SAFETY_ERR_MASK is shown in Table 10-260.

Return to the Summary Table.

Table 10-260. HWA_SAFETY_ERR_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU1	R	0h	
9	HWA_SAFETY_ACCESS_ERR_MASK_OPONG_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM3 access error is masked.1'b 0 : ACCEL_MEM3 access error is not masked
8	HWA_SAFETY_ACCESS_ERR_MASK_OPING_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM2 access error is masked.1'b 0 : ACCEL_MEM2 access error is not masked
7	HWA_SAFETY_ACCESS_ERR_MASK_IPONG_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM1 access error is masked.1'b 0 : ACCEL_MEM1 access error is not masked
6	HWA_SAFETY_ACCESS_ERR_MASK_IPING_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM0 access error is masked.1'b 0 : ACCEL_MEM0 access error is not masked
5	HWA_SAFETY_PARITY_ERR_MASK_OPONG_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM3 parity error is masked.1'b 0 : ACCEL_MEM03 parity error is not masked
4	HWA_SAFETY_PARITY_ERR_MASK_OPING_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM2 parity error is masked.1'b 0 : ACCEL_MEM2 parity error is not masked
3	HWA_SAFETY_PARITY_ERR_MASK_IPONG_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM1 parity error is masked.1'b 0 : ACCEL_MEM1 parity error is not masked
2	HWA_SAFETY_PARITY_ERR_MASK_IPING_RAM	R/W	0h	When 1'b 1 : ACCEL_MEM0 parity error is masked.1'b 0 : ACCEL_MEM0 parity error is not masked
1	HWA_SAFETY_PARITY_ERR_MASK_WINDOW_RAM	R/W	0h	When 1'b 1 : Window RAM parity error is masked.1'b 0 : Window RAM parity error is not masked
0	HWA_SAFETY_ERR_MASK_FSM_LOCKSTEP	R/W	0h	When 1'b 1 : FSM lockstep error is masked.1'b 0 : FSM lockstep error is not masked

10.4.245 HWA_SAFETY_ERR_STATUS Register (Offset = 3D0h) [Reset = 00000000h]

HWA_SAFETY_ERR_STATUS is shown in Table 10-261.

Return to the Summary Table.

Table 10-261. HWA_SAFETY_ERR_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU1	R	0h	
9	HWA_SAFETY_ACCESS_ERR_STATUS_OPONG_RAM	R	0h	Indicates the ACCEL_MEM3 access error (Masked status)
8	HWA_SAFETY_ACCESS_ERR_STATUS_OPING_RAM	R	0h	Indicates the ACCEL_MEM2 access error (Masked status)
7	HWA_SAFETY_ACCESS_ERR_STATUS_IPONG_RAM	R	0h	Indicates the ACCEL_MEM1 access error (Masked status)
6	HWA_SAFETY_ACCESS_ERR_STATUS_IPING_RAM	R	0h	Indicates the ACCEL_MEM0 access error (Masked status)

Table 10-261. HWA_SAFETY_ERR_STATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	HWA_SAFETY_PARITY_ERR_STATUS_OPONG_RAM	R	0h	Indicates the ACCEL_MEM3 parity error (Masked status)
4	HWA_SAFETY_PARITY_ERR_STATUS_OPING_RAM	R	0h	Indicates the ACCEL_MEM2 parity error (Masked status)
3	HWA_SAFETY_PARITY_ERR_STATUS_IPONG_RAM	R	0h	Indicates the ACCEL_MEM1 parity error (Masked status)
2	HWA_SAFETY_PARITY_ERR_STATUS_IPING_RAM	R	0h	Indicates the ACCEL_MEM0 parity error (Masked status)
1	HWA_SAFETY_PARITY_ERR_STATUS_WINDOW_RAM	R	0h	Indicates the Window RAM parity error (Masked status)
0	HWA_SAFETY_ERR_STATUS_FSM_LOCKSTEP	R	0h	Indicates the FSM lockstep error (Masked status)

10.4.246 HWA_SAFETY_ERR_STATUS_RAW Register (Offset = 3D4h) [Reset = 00000000h]

HWA_SAFETY_ERR_STATUS_RAW is shown in Table 10-262.

Return to the Summary Table.

Table 10-262. HWA_SAFETY_ERR_STATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU1	R	0h	
9	HWA_SAFETY_ACCESS_ERR_STATUS_RAW_OPONG_RAM	R	0h	Indicates the ACCEL_MEM3 access error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 9
8	HWA_SAFETY_ACCESS_ERR_STATUS_RAW_OPING_RAM	R	0h	Indicates the ACCEL_MEM2 access error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 8
7	HWA_SAFETY_ACCESS_ERR_STATUS_RAW_IPONG_RAM	R	0h	Indicates the ACCEL_MEM1 access error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 7
6	HWA_SAFETY_ACCESS_ERR_STATUS_RAW_IPING_RAM	R	0h	Indicates the ACCEL_MEM0 access error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 6
5	HWA_SAFETY_PARITY_ERR_STATUS_RAW_OPONG_RAM	R	0h	Indicates the ACCEL_MEM3 parity error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 5
4	HWA_SAFETY_PARITY_ERR_STATUS_RAW_OPING_RAM	R	0h	Indicates the ACCEL_MEM2 parity error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 4
3	HWA_SAFETY_PARITY_ERR_STATUS_RAW_IPONG_RAM	R	0h	Indicates the ACCEL_MEM1 parity error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 3
2	HWA_SAFETY_PARITY_ERR_STATUS_RAW_IPING_RAM	R	0h	Indicates the ACCEL_MEM0 parity error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 2
1	HWA_SAFETY_PARITY_ERR_STATUS_RAW_WINDOW_RAM	R	0h	Indicates the Window RAM parity error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 1
0	HWA_SAFETY_ERR_STATUS_RAW_FSM_LOCKSTEP	R	0h	Indicates the FSM lockstep error (raw status).Set irrespective of HWA_SAFETY_ERR_MASK bit 0

11 Enhanced Direct Memory Access (EDMA)

This section describes the Enhanced Direct Memory Access (EDMA) controller. 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 follower 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 EDMA 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.

11.1 EDMA Module Overview

The enhanced direct memory access module, also called EDMA, performs high-performance data transfers between two target endpoints, 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 11-1 shows an overview of the EDMA module.

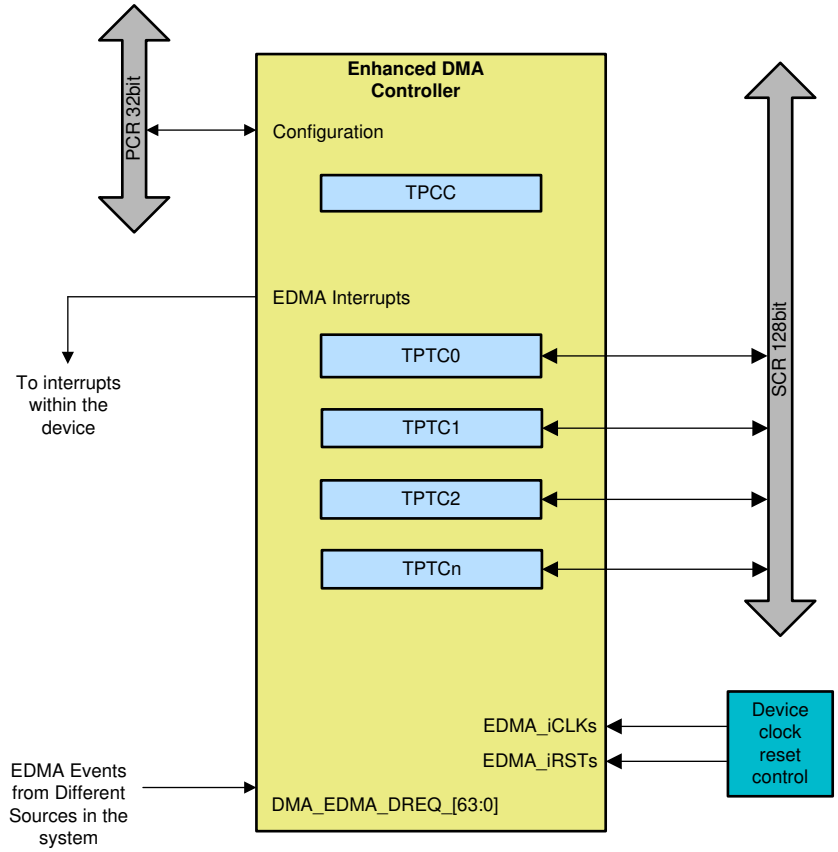


Figure 11-1. EDMA Module Overview

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 target 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.

11.1.1 EDMA Features

This section shows generic EDMA features. For features applicable to the EDMA instances in the device, 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.
- 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

11.2 EDMA Integration

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

11.2.1 EDMA Integration

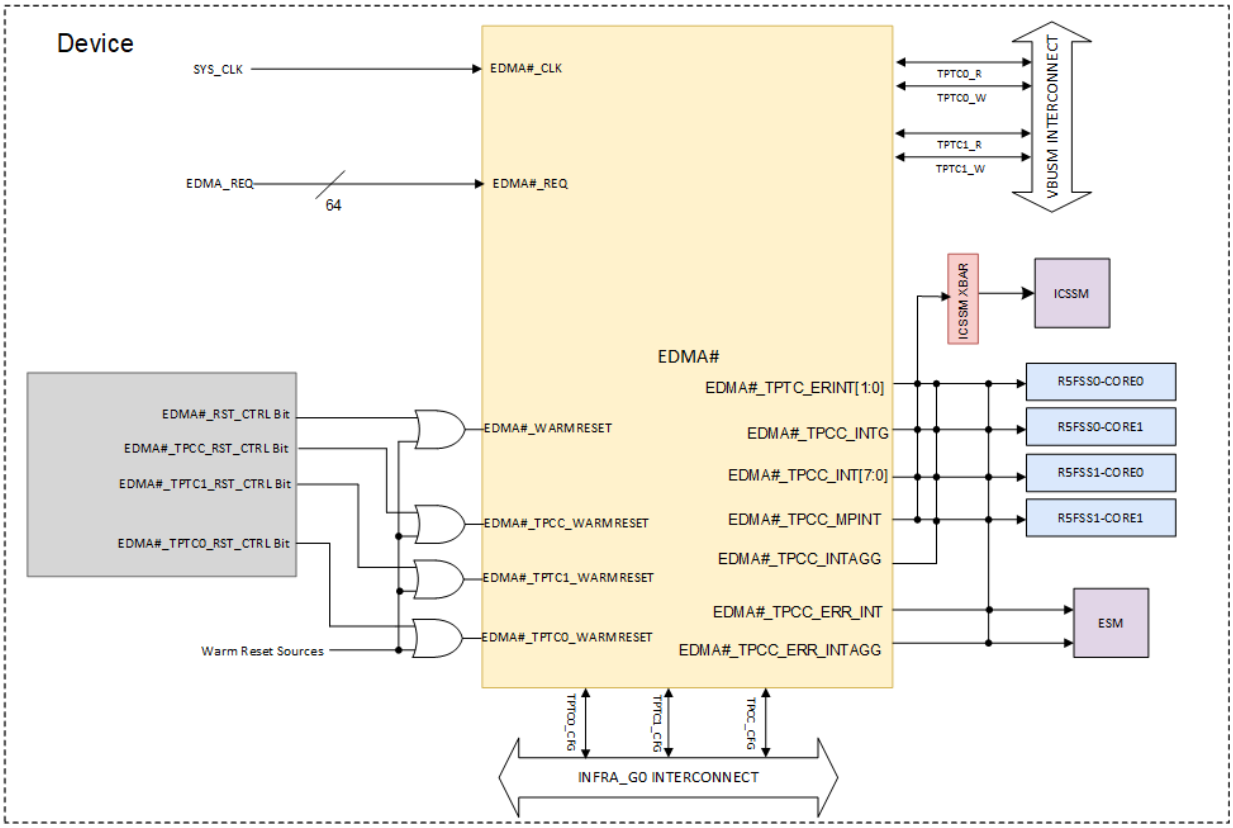


Figure 11-2. EDMA Integration Block Diagram

Note

For more information on the interconnects, see the *System Interconnect* chapter.

For more information on power, reset, and clock management, see the corresponding sections within the *Device Configuration* chapter.

For more information on the device interrupt controllers, see the *Interrupt Controllers* chapter.

11.2.2 EDMA Interrupt Aggregator

The following EDMA interrupts are aggregated and sent to the processor:

- TPCC Completion Interrupt
- TPCC Completion Region Interrupts
- TPTCs Completion Interrupt

The table below shows the associated interrupt and registers for each TPCC instance.

TPCC	Interrupt	Registers
TPCC_A	APPSS_TPCC_A_ERRAGG	APP_CTRL: APPSS_TPCC_A_INTAGG_MASK APP_CTRL: APPSS_TPCC_A_INTAGG_STATUS APP_CTRL: APPSS_TPCC_A_INTAGG_STATUS_RAW
TPCC_B	APPSS_TPCC_B_ERRAGG	APP_CTRL: APPSS_TPCC_B_INTAGG_MASK APP_CTRL: APPSS_TPCC_B_INTAGG_STATUS APP_CTRL: APPSS_TPCC_B_INTAGG_STATUS_RAW

For an event to generate an interrupt to the processor, the corresponding bit field must be unmasked in TPCC_x_INTAGG_MASK.

Only an interrupt processor can read the TPCC_x_INTAGG_STATUS register to detect which event triggered the interrupt.

The interrupt can be cleared by writing 0x1 to the corresponding bit in TPCC_x_INTAGG_STATUS.

The software must verfiy that all the aggregated interrupts are cleared so that the level interrupt is de-asserted before exiting the ISR. Only then the software can provide a new pulse interrupt to the processor. Thus, after clearing the software can read the register to confirm a value of 0x0.

The register TPCC_x_INTAGG_STATUS_RAW is set on an event irrespective of the value in TPCC_x_INTAGG_MASK. This field can be cleared by writing 0x1 to the corresponding bit in TPCC_x_INTAGG_STATUS_RAW.

11.2.3 EDMA Error Interrupt Aggregator

The following interrupts are aggregated and sent to the processor

- TPCC Error
- TPCC MPU Error
- TPTCs Error
- TPCC Read and Write Config Space Access error
- TPTCs Read and Write Config Space Access error

The table below lists the associated interrupt and registers for each TPCC instance.

TPCC	Interrupt	Registers
TPCC_A	APPSS_TPCC_A_ERRAGG	APP_CTRL: APPSS_TPCC_A_ERRAGG_MASK APP_CTRL: APPSS_TPCC_A_ERRAGG_STATUS APP_CTRL: APPSS_TPCC_A_ERRAGG_STATUS_RAW
TPCC_B	APPSS_TPCC_B_ERRAGG	APP_CTRL: APPSS_TPCC_B_ERRAGG_MASK APP_CTRL: APPSS_TPCC_B_ERRAGG_STATUS APP_CTRL: APPSS_TPCC_B_ERRAGG_STATUS_RAW

For an event to generate an interrupt to the processor, the corresponding bit field must be unmasked in TPCC_x_ERRAGG_MASK. On an interrupt processor can read the TPCC_x_ERRAGG_STATUS register to detect which event triggered the interrupt. The interrupt can be cleared by writing 0x1 to the corresponding bit in TPCC_x_ERRAGG_STATUS. It is the SW responsibility to ensure that all the aggregated interrupts are cleared so that the level interrupt is de-asserted before exiting the ISR. Only then is it guaranteed that a new pulse interrupt will be generated to the processor. Hence after clearing SW should read the register to confirm a value of 0x0 The register TPCC_x_ERRAGG_STATUS_RAW is set on an event irrespective of the value in TPCC_x_ERRAGG_MASK. This field can be cleared by writing 0x1 to the corresponding bit in TPCC_x_ERRAGG_STATUS_RAW.

11.2.4 EDMA Configuration

The table below lists the configuration registers for TPTCs.

TPTC_A0	APP_CTRL:APPSS_TPTC_BOUNDARY_CFG. APPSS_TPTC_BOUNDARY_CFG_TPTC_A0_SIZE APP_CTRL:APPSS_TPTC_XID_REORDER_CFG:APPSS_TPTC_XID_REORDER_CFG_TPTC_A0_DISABLE
TPTC_A1	APP_CTRL:APPSS_TPTC_BOUNDARY_CFG. APPSS_TPTC_BOUNDARY_CFG_TPTC_A1_SIZE APP_CTRL:APPSS_TPTC_XID_REORDER_CFG:APPSS_TPTC_XID_REORDER_CFG_TPTC_A1_DISABLE

TPTC_B0	APP_CTRL:APPSS_TPTC_BOUNDARY_CFG. APPSS_TPTC_BOUNDARY_CFG_TPTC_B0_SIZE APP_CTRL:APPSS_TPTC_XID_REORDER_CFG:APPSS_TPTC_XID_REORDER_CFG_TPTC_B0_DISABLE
TPTC_B1	APP_CTRL:APPSS_TPTC_BOUNDARY_CFG. APPSS_TPTC_BOUNDARY_CFG_TPTC_B1_SIZE APP_CTRL:APPSS_TPTC_XID_REORDER_CFG:APPSS_TPTC_XID_REORDER_CFG_TPTC_B1_DISABLE

The APPSS has 1 TPCC (TPCC_A) and 2 TPTCs. The HWASS has 1 TPCC (TPCC_B) and 2 TPTCs.

The table below lists the **TPTC** configurations in APPSS/HWASS.

PARAMETER	**APPSS TPTC 1**	**APPSS TPTC 2**	**HWASS Both TPTC**
FIFO Size	256B	64B	64B
FIFO Type + Safety	Flops with ECC	Flops with ECC	Flops with ECC
TR Pipe Depth	2	2	2
Bus Width	8B	8B	8B
Read Cmd Num	8	8	8
Write Cmd Num	8	8	8
Support MAX DBS (Default Burst Size)	64B	32B (half of FIFO Size)	32B (half of FIFO Size)
Default DBS	64B	32B	32B

The table below lists the **TPCC** configurations in APPSS/HWASS

Parameters	**APPSS TPCC**	**HWASS TPCC**
DMA Ch	64	64
Param Entires	128	128
QDMA Ch	8	8
Event queues	2	2
Mem Protection	Yes	Yes
Channel Mapping	Yes	Yes
Num TCs	2	2
Num Int Ch	64	64
Num Regions	8	8

11.3 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 integration section of the device to determine the applicability of these features.

11.3.1 Block Diagram

Figure 11-3 shows the functional block diagram of the EDMA controller.

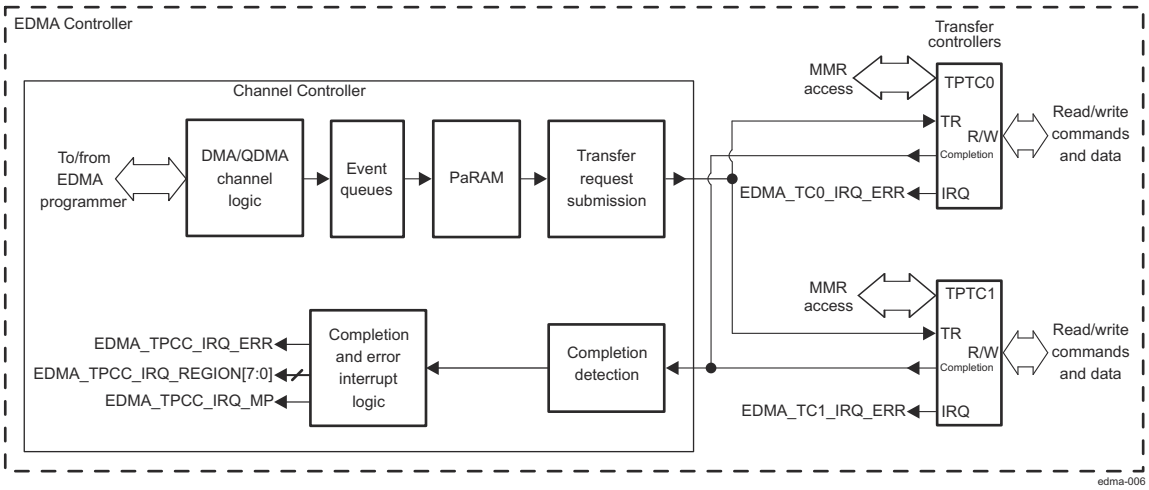


Figure 11-3. EDMA Controller Block Diagram

11.3.1.1 Third-Party Channel Controller

The TPCC is the EDMA transfer scheduler responsible for scheduling, arbitrating, and issuing user programmed transfers to the two TPTCs.

The functional block diagram below describes EDMA channel controller (EDMA_TPCC).

A. Although the block is depicted twice in EDMA Channel Controller Block Diagram, there is only one physical register set for the QDMA to PaRAM set mapping block.

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 follower 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.

The EDMA_TPCC supports up to 64 DMA channels and up to 8 QDMA channels. These channels are identical, except for the triggers:

- 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.).

11.3.1.2 Third-Party Transfer Controller

The TPTC module is the EDMA transfer engine that generates transfers as programmed in dedicated working registers, using two dedicated controller ports: a read-only port and a write-only port.

Figure 11-4 shows a functional block diagram and of the EDMA transfer controller (EDMA_TPTC) and its connection to the EDMA_TPCC.

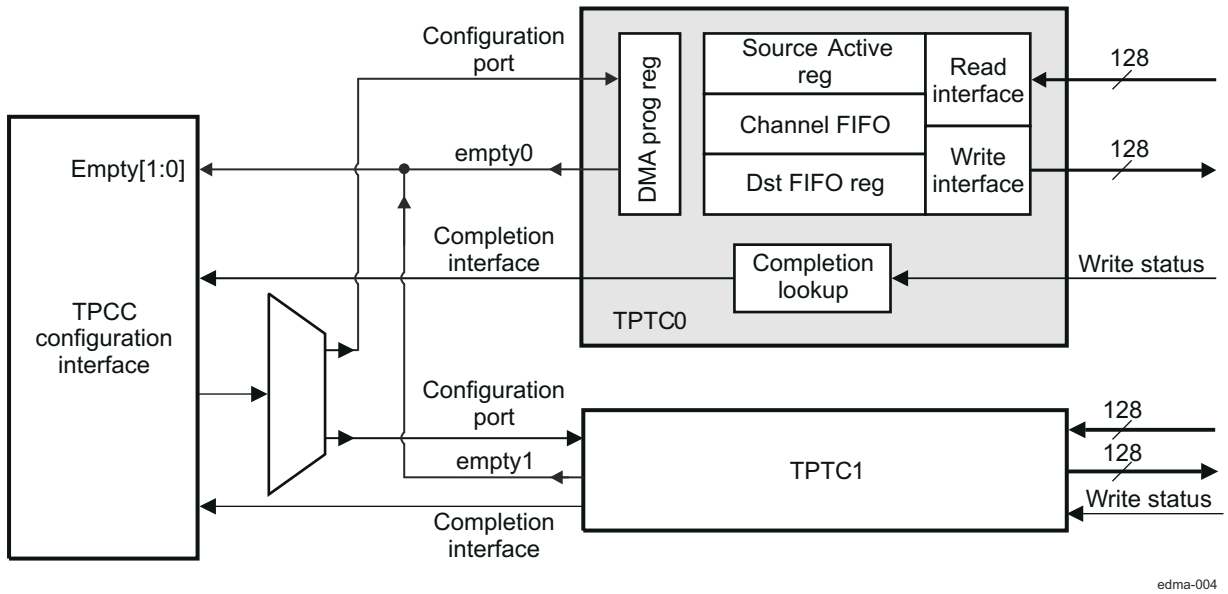


Figure 11-4. TPTC Block Diagram

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: Target 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.

11.3.2 Types of EDMA controller Transfers

An EDMA transfer is always defined in terms of three dimensions. Figure 11-5 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.

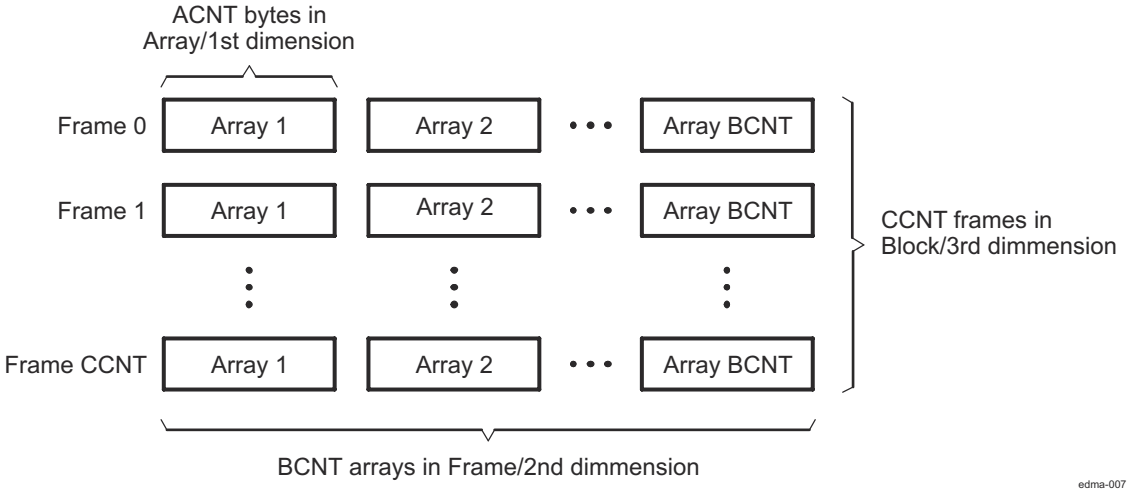


Figure 11-5. Definition of ACNT, BCNT, and CCNT

11.3.2.1 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 11-6, 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 11-6, SRCCIDX / DSTCIDX is the difference between the start of Frame 0 Array 3 to the start of Frame 1 Array 0.

Figure 11-6 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 11-6 for details on parameter set updates.

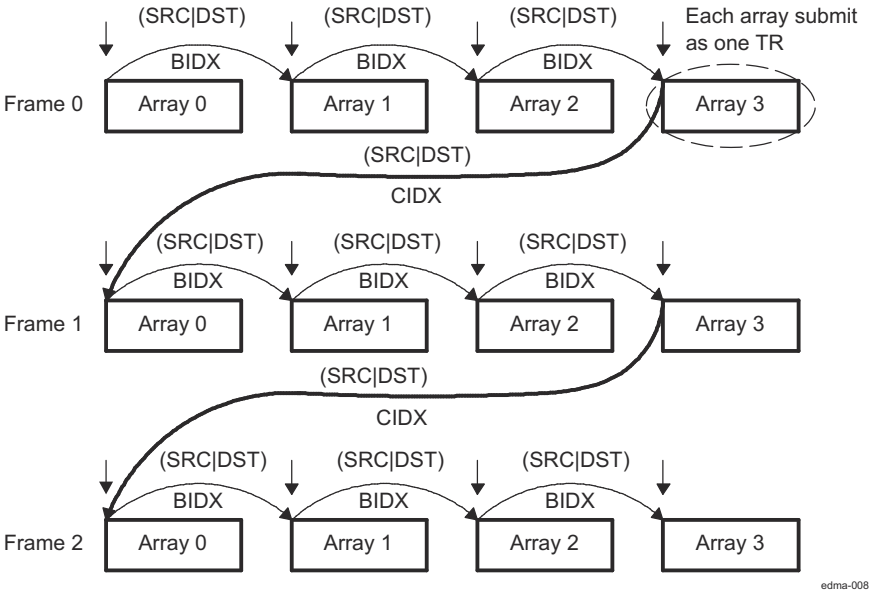


Figure 11-6. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)

ADVANCE INFORMATION

11.3.2.2 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 11-7. 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 11.3.3.6 for details on parameter set updates.

Figure 11-7 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.

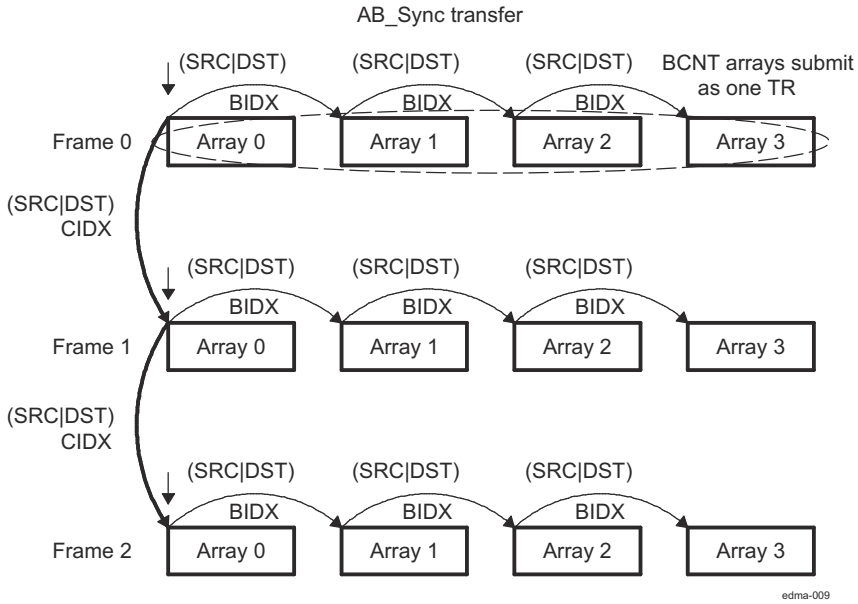


Figure 11-7. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)

Note

ABC-synchronized transfers are not directly supported. It can be logically achieved by chaining between multiple AB-synchronized transfers.

11.3.3 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:

- PaRAM sets
- 64 channels that are direct mapped and can be used as link or QDMA sets if not used for DMA channels

- 8 channels remain for link or QDMA sets

By default, all channels map to PaRAM set to 0 and should be remapped before use by EDMA_TPCC_DCHMAPN_m and EDMA_TPCC_QCHMAPN_j registers.

Table 11-1. EDMA Parameter RAM Contents

PaRAM Set Number	Base Address	Parameters
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
...	...	...
127	EDMA Base Address + 5000h to EDMA Base Address + 4FE0h	PaRAM set 127

Note

11.3.3.1 PaRAM

Each parameter set of PaRAM is organized into eight 32-bit words or 32 bytes, as shown in PaRAM Set and described in EDMA Channel Parameter Description. Each PaRAM set consists of 16-bit and 32-bit parameters.

Figure 11-8. PaRAM Set

Note

Figure above is a representation of 128 bit entries. For device specific details please refer to Section 11.2.4 chapter.

ADVANCE INFORMATION

Table 11-2. 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	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	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	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	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 cam result in undefined behavior.

11.3.3.2 EDMA Channel PaRAM Set Entry Fields

11.3.3.2.1 Channel Options Parameter (OPT)

This is the control register for TPCC channel configuration options. Refer to the EDMA_TPCC_OPT_n register bitfield description for additional details.

11.3.3.2.2 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 *Error Generation* for additional details.

11.3.3.2.3 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 *Error Generation* for additional details.

11.3.3.2.4 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 11.3.3.5 *Dummy Versus Null Transfer Comparison* and Section 11.3.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

11.3.3.2.5 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 11.3.3.5 *Dummy Versus Null Transfer Comparison* and Section 11.3.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

11.3.3.2.6 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 11.3.3.5 *Dummy Versus Null Transfer Comparison* and Section 11.3.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

11.3.3.2.7 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.

11.3.3.2.8 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.

11.3.3.2.9 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 11.3.3.2.8 *Source B Index (SBIDX)* for examples.

11.3.3.2.10 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 11-6), while the current array in an AB-synchronized transfer is the first array in the frame (Figure 11-7).

11.3.3.2.11 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 11-6), while the current array in a AB-synchronized transfer is the first array in the frame (Figure 11-7).

11.3.3.2.12 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 11.3.5 *Completion of a DMA Transfer* for details on terminating a transfer.

11.3.3.3 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.

11.3.3.4 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.

11.3.3.5 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 11-3 summarizes the conditions and effects of null and dummy transfer requests.

Table 11-3. 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

11.3.3.6 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 AB-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 for details and conditions on the parameter updates. A link update occurs when the PaRAM set is exhausted, as described in Section 11.3.3.8 *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 *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.

shows the details of parameter updates that occur within EDMA_TPCC for A-synchronized and AB-synchronized transfers.

Table 11-4. 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

Table 11-4. Parameter Updates in EDMA_TPCC (for Non-Null, Non-Dummy PaRAM Set) (continued)

	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
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	None	None	= LINK.EDMA_TPCC_OPT_n	None	None	= LINK.EDMA_TPCC_OPT_n

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 target endpoint.

11.3.3.7 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 *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, target 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.

11.3.3.8 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 11.3.3.3 *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 11-9 shows an example of a linked transfer.

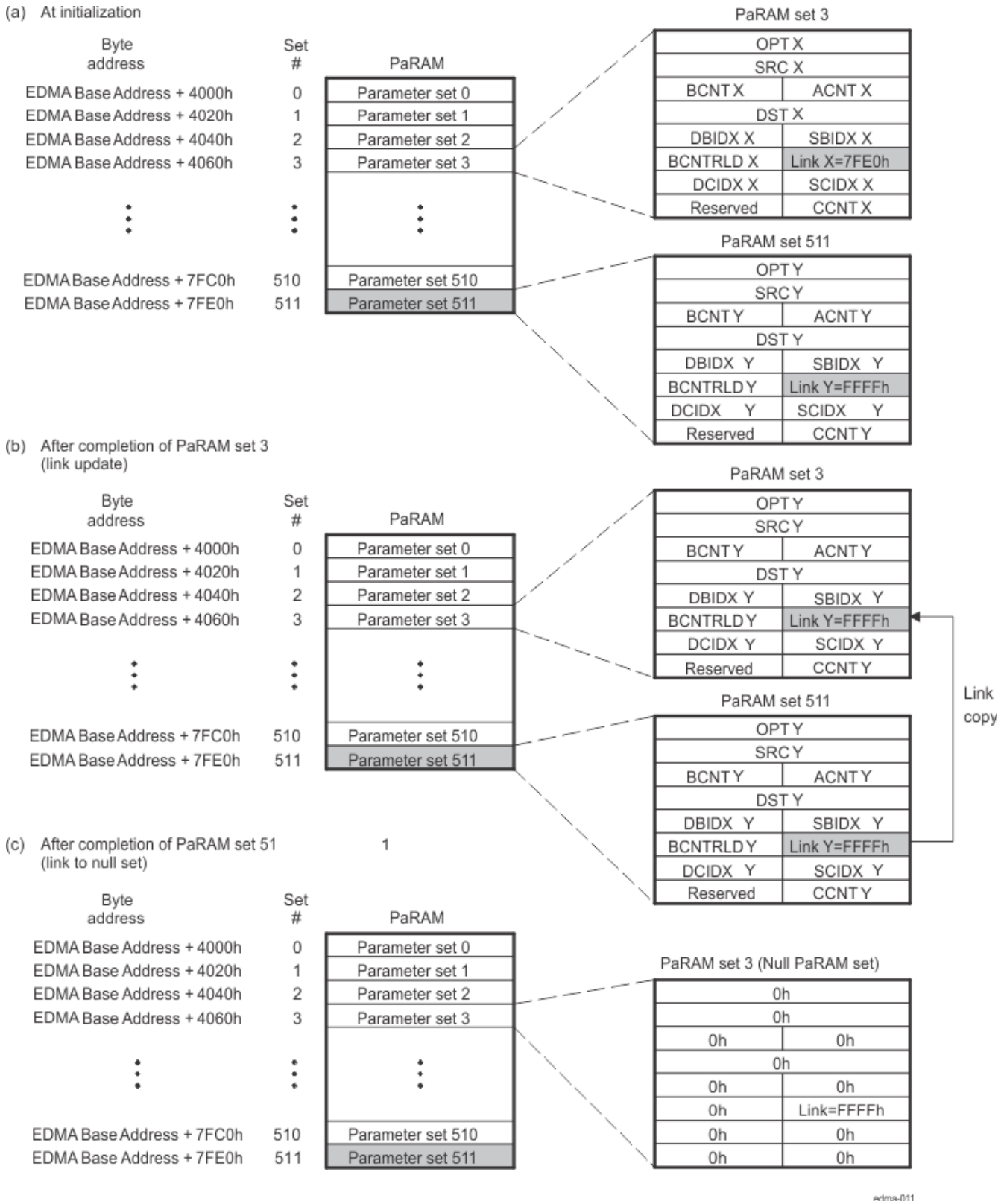


Figure 11-9. Linked Transfer

Note

AM273x has a maximum of 128 PaRAM sets. Additional tables and diagrams in this chapter may show a larger number (up to 511), however 128 is the maximum allowed number of entries.

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 11.3.6 *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 11.3.4.2 *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 11-10 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).

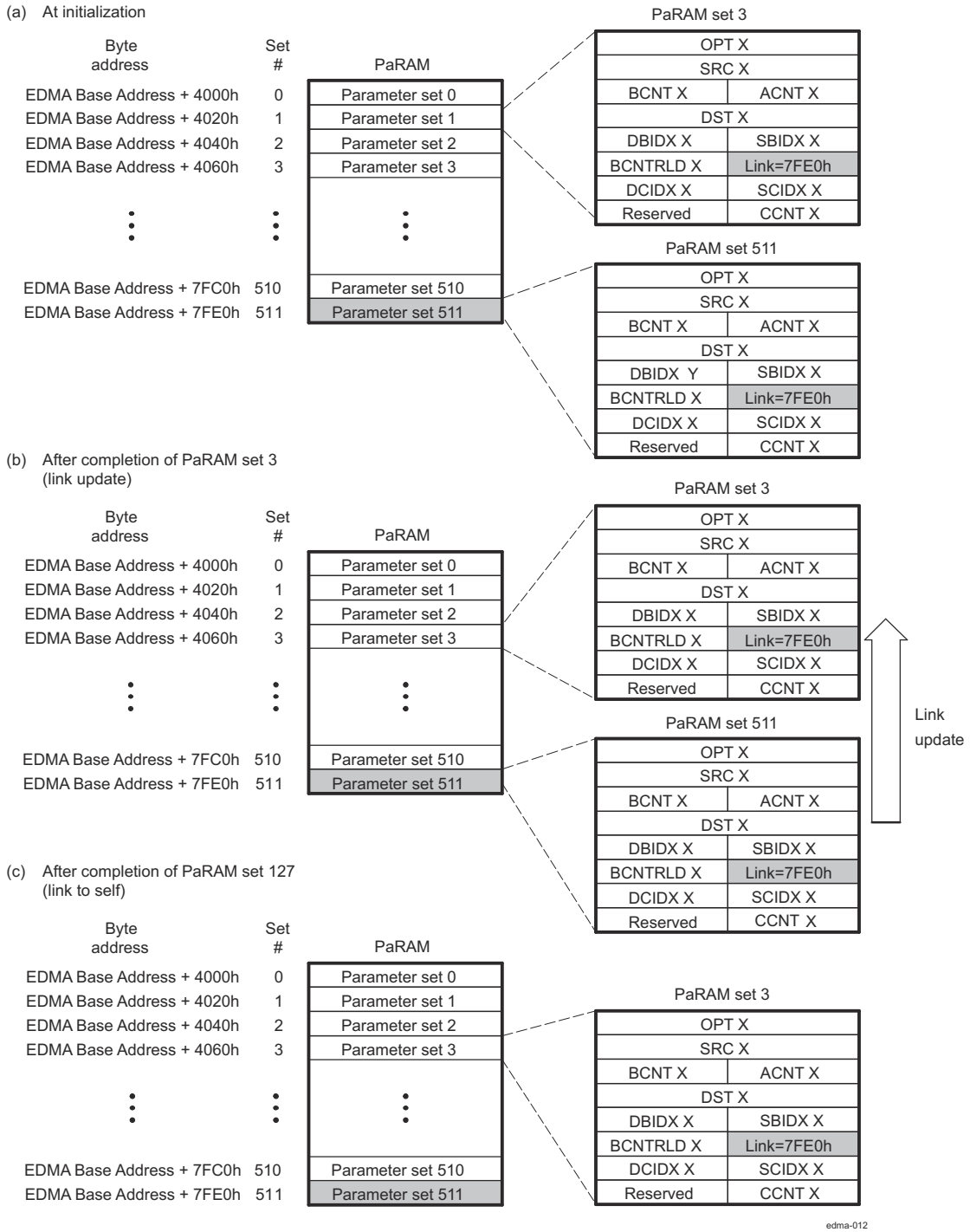


Figure 11-10. Link-to-Self Transfer

Note

If the in EDMA_TPCC_OPT_n[3] STATIC bit is set for a PaRAM set, then link updates are not performed.

11.3.3.9 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.

11.3.4 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.

11.3.4.1 DMA Channels

11.3.4.1.1 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 11.3.3.3 *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).

11.3.4.1.2 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 11.3.3.3 *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.

11.3.4.1.3 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 11.3.3.3 *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).

11.3.4.2 QDMA Channels

11.3.4.2.1 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.

11.3.4.3 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.

11.3.5 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 11-5 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 11-5. 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

ADVANCE INFORMATION

Table 11-5. Expected Number of Transfers for Non-Null Transfer (continued)

Sync Mode	Counts at time 0	Total # Transfers	Counts prior to final TR
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 11.3.8 *Chaining EDMA Channels* for details on chaining (intermediate/final chaining) and Section 11.3.9 *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 11.3.3.8 *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 11.3.3.5 *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.

11.3.5.1 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.

11.3.5.2 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.

11.3.5.3 Dummy or Null Completion

This is a variation of early completion. Dummy or null completion is associated with a dummy set Section 11.3.3.4 or null set Section 11.3.3.3. 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).

11.3.6 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.

11.3.6.1 DMA Channel to PaRAM Mapping

The mapping between the DMA channel numbers and the PaRAM sets is programmable (see). 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. illustrates the use of EDMA_TPCC_DCHMAPN_m. There is one EDMA_TPCC_DCHMAPN_m register per channel.

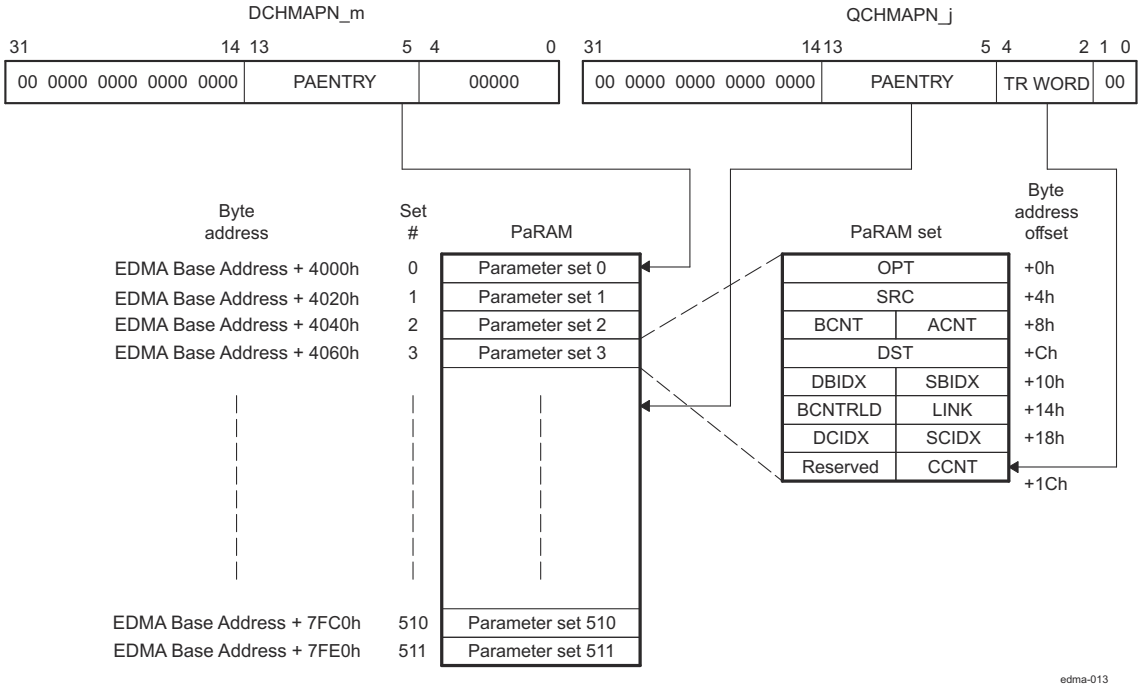


Figure 11-11. DMA Channel and QDMA Channel to PaRAM Mapping

Note

ADVANCE INFORMATION

11.3.6.2 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 11-12 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.

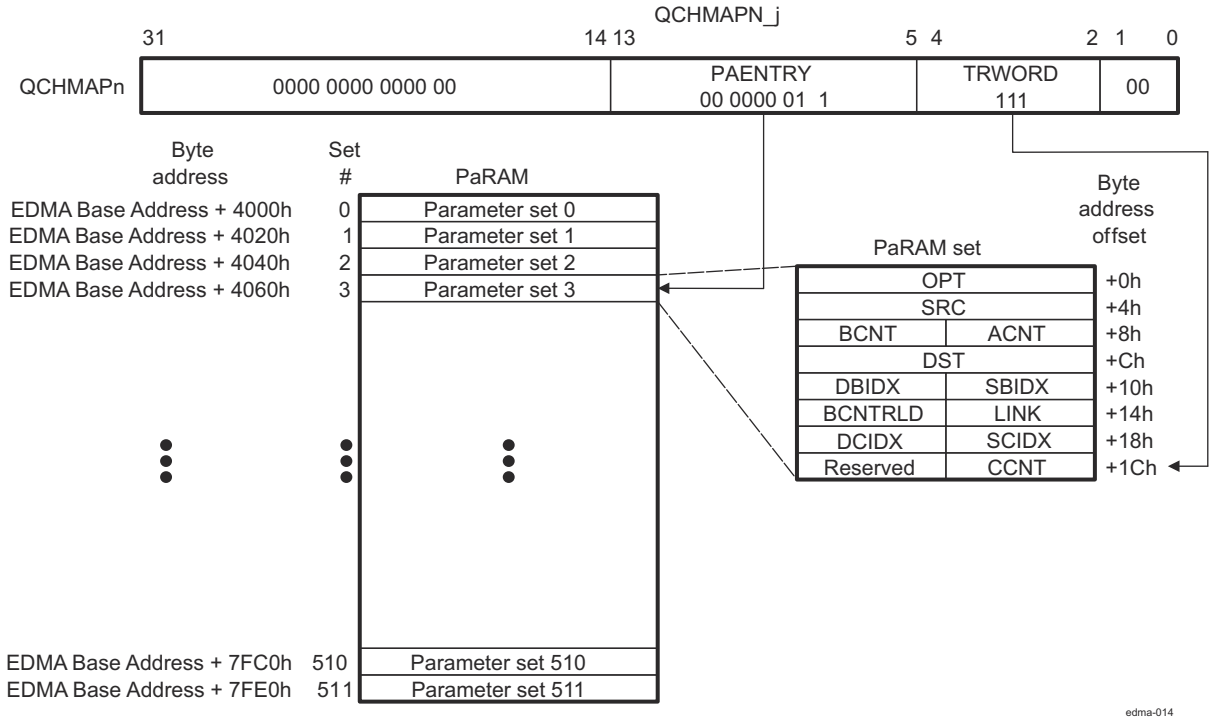


Figure 11-12. QDMA Channel to PaRAM Mapping

11.3.7 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 11.3.10 *Memory Protection*.

11.3.7.1 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 11-6 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 11-6. 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 11-13 illustrates the conceptual view of the regions.

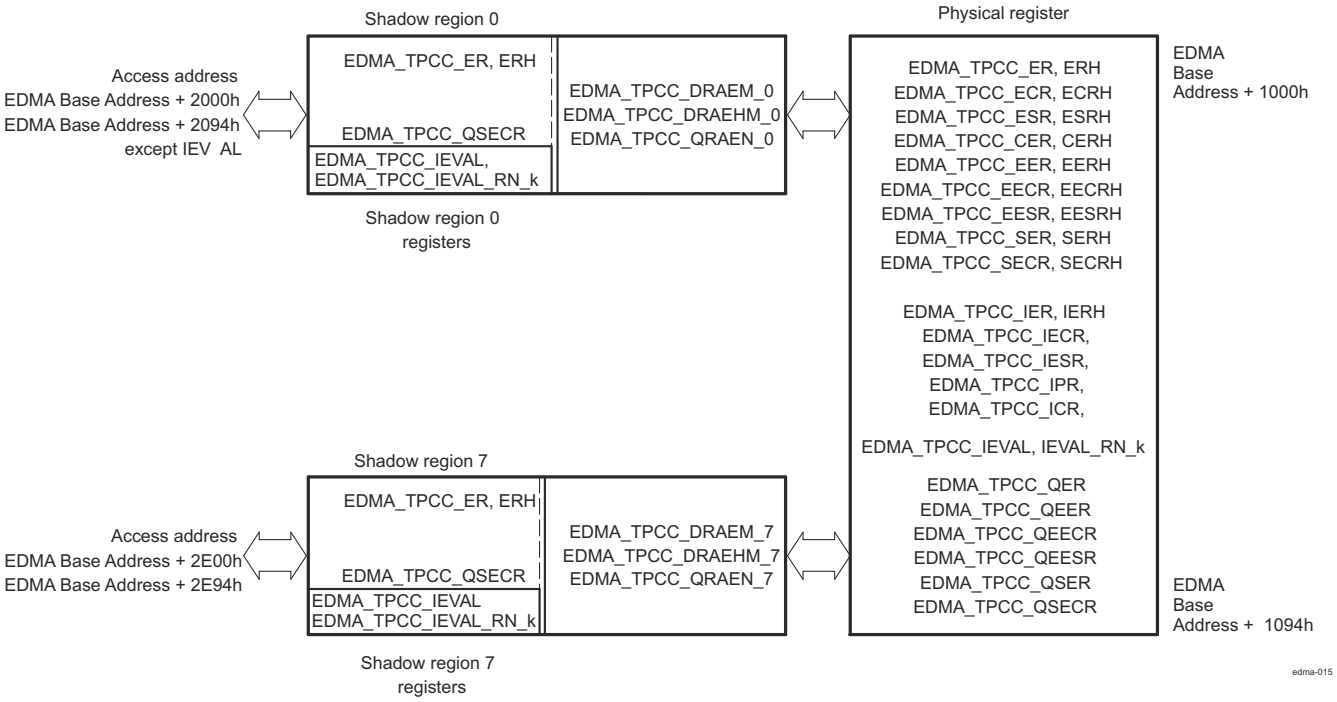


Figure 11-13. Shadow Region Registers

11.3.7.2 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 (8 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 11-1. 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
```

11.3.7.3 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.

11.3.8 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 11.3.3.8 *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 11.3.4.1.3 *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 11-7 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 11-7. 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)

11.3.9 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 11-8. The transfer completion interrupts and the error interrupts from the transfer controllers are all routed to the device interrupt controllers INTCs.

Table 11-8. 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 11-9. 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

11.3.9.1 Transfer Completion Interrupts

The EDMA_TPCC is responsible for generating transfer completion interrupts to the CPU(s) (and other EDMA controllers). 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 11.3.9.1.1 *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 11-10. 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 11-11 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 11-11. 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)

11.3.9.1.1 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 11-14

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 11-14 shows the Interrupt diagram of the EDMA controller.

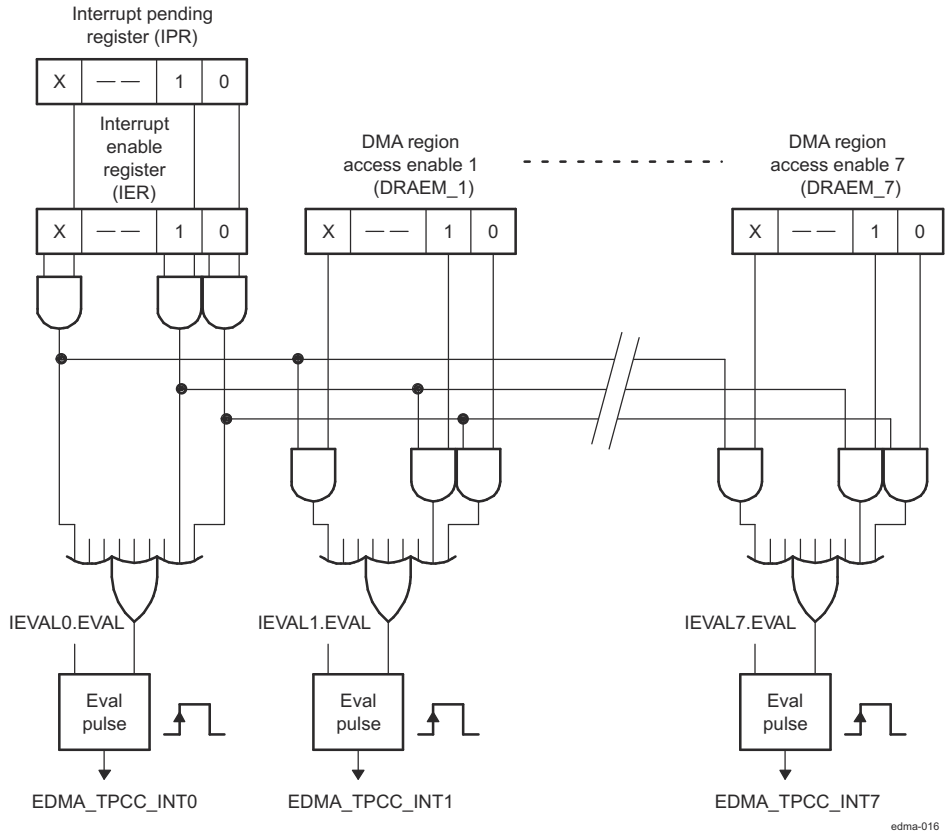


Figure 11-14. Interrupt Diagram

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)

ADVANCE INFORMATION

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).

11.3.9.1.2 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.

11.3.9.2 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 11-2 and Example 11-3.

The ISR routine in Example 11-2 is more exhaustive and incurs a higher latency.

Example 11-2. 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).
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 11-3. 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.

11.3.9.3 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 controller associated with the shadow region) if the software architecture chooses not to use all interrupts. Refer to Example 11-3 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 11.3.9.4 *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.

ADVANCE INFORMATION

11.3.9.4 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 11-15 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.

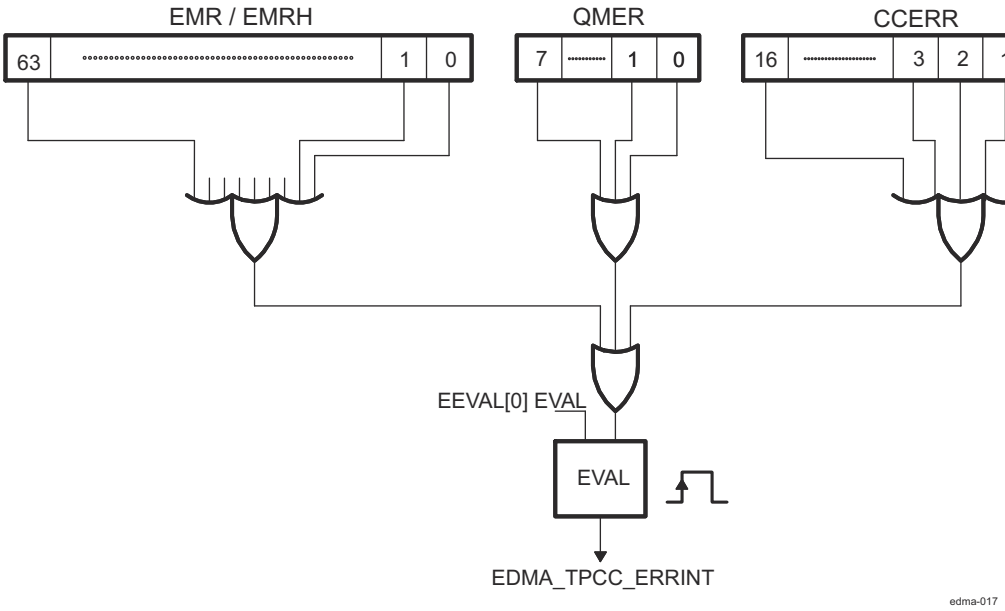


Figure 11-15. Error Interrupt Operation

11.3.10 Memory Protection

The EDMA channel controller supports two kinds of memory protection: active and proxy.

11.3.10.1 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 the associated Register Addendum.

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 11-12 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 11-12. Allowed Accesses

Access	Supervisor	User
Read	Yes	Yes
Write	Yes	No

Table 11-13 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 11-13. 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 11-14. 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.

ADVANCE INFORMATION

Table 11-15. 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.

11.3.10.2 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 11-16.

Figure 11-16. 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)

(b) Channel Options Parameter (OPT_n) Content

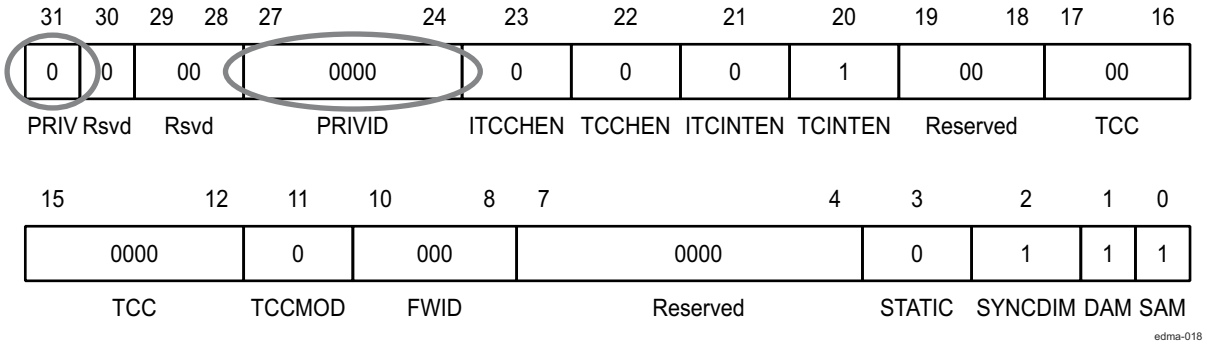


Figure 11-17. Channel Options Parameter (OPT) Example

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 11-18 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 target memories).

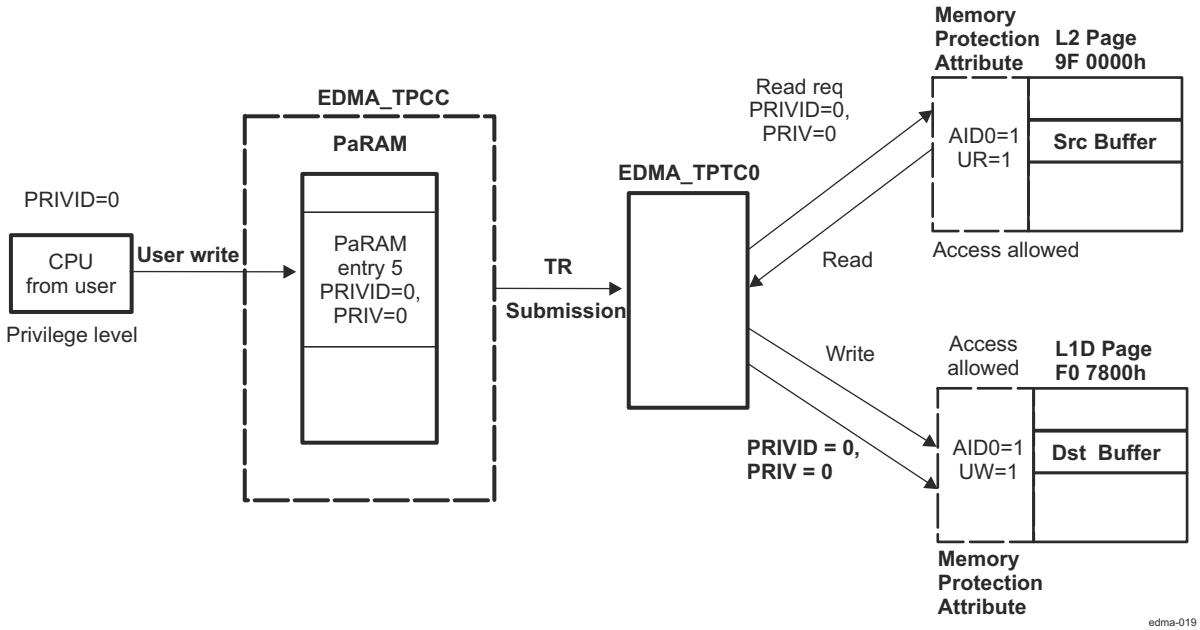


Figure 11-18. Proxy Memory Protection Example

ADVANCE INFORMATION

11.3.11 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 *Performance Considerations* 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 the associated Register Addendum for EDMA_TPCC_Q0E_p / EDMA_TPCC_Q1E_p descriptions of the bit fields.

11.3.11.1 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 *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.

11.3.11.2 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.

11.3.11.3 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.

11.3.11.4 Performance Considerations

The device system bus infrastructure arbitrates bus requests from all of the controllers (TCs, CPU(S), and other bus controllers) to the shared target resources (peripherals and memories).

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.

11.3.12 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 target to the EDMA_TPCC.

11.3.13 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 target 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 target endpoint.
10. This continues until the TR completes and the EDMA_TPTC*n* then signals completion status to the EDMA_TPCC.

11.3.14 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 11-19 shows the different places EDMA priorities come into play.

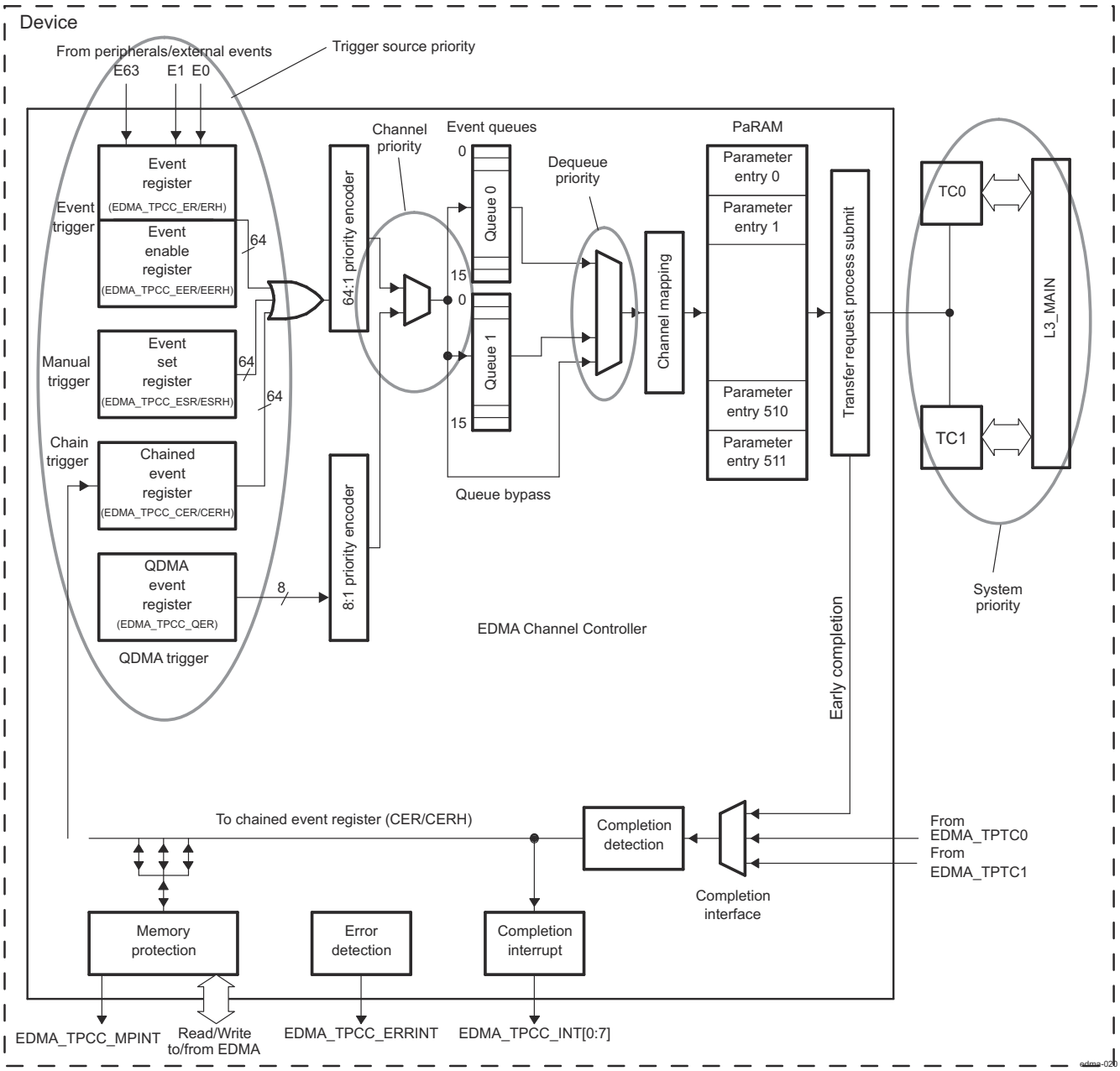


Figure 11-19. EDMA Prioritization

11.3.14.1 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.

11.3.14.2 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.

11.3.14.3 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.

11.3.15 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 controller and target 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.

11.4 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.

11.4.1 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 11-20.

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 11-21 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 11-21 shows the parameters Block Move transfer.

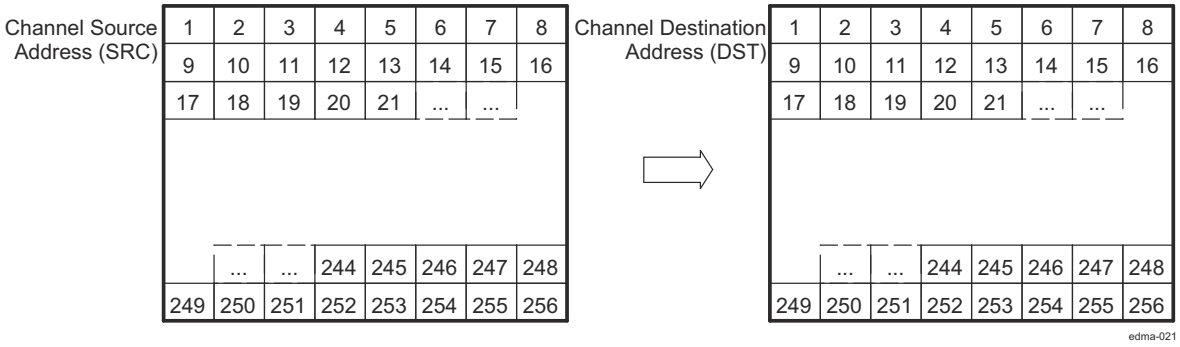


Figure 11-20. Block Move Example

ADVANCE INFORMATION

Figure 11-21. Block Move Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0008h		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
0001h	0100h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0000h	0000h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	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

11.4.2 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 11-22 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 11-23 shows the parameters for Subframe Extraction transfer.

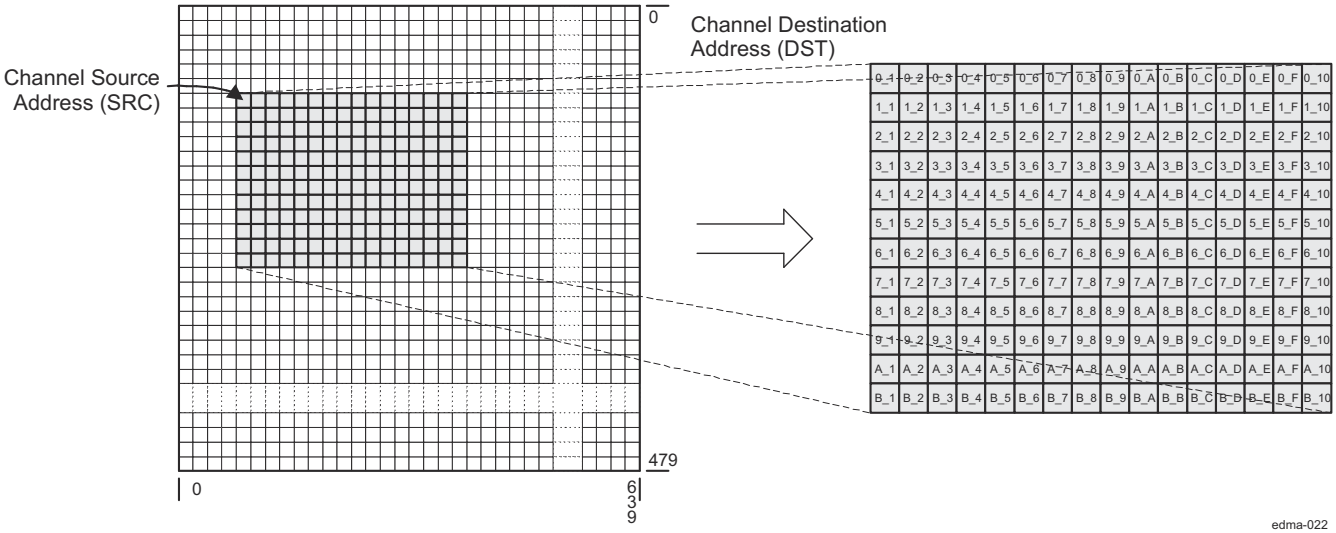


Figure 11-22. Subframe Extraction Transfer

Figure 11-23. Subframe Extraction Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents			Parameter	
0010 000Ch			Channel Options Parameter (OPT)	
Channel Source Address (SRC)			Channel Source Address (SRC)	
000Ch	0020h		Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)			Channel Destination Address (DST)	
0020h	0500h		Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh		BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h		Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h		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

11.4.3 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 11-25 shows the parameter set programming for this transfer, assuming channel 0 and an element size of 4 bytes.

Figure 11-24 shows the Data Sorting transfer

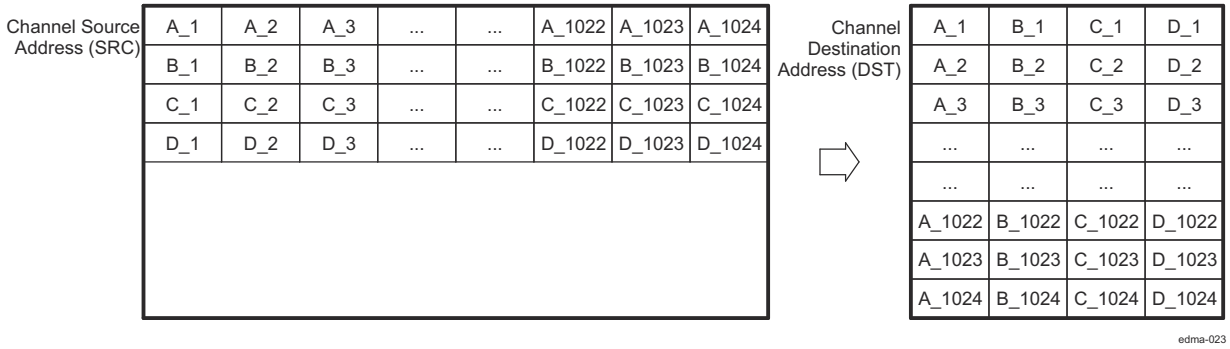


Figure 11-24. Data Sorting Example

Figure 11-25. Data Sorting Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents			Parameter	
0090 0004h			Channel Options Parameter (OPT)	
Channel Source Address (SRC)			Channel Source Address (SRC)	
0400h	0004h		Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)			Channel Destination Address (DST)	
0010h	0001h		Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh		BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h		Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0004h		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

11.4.4 Setting Up an EDMA Transfer

The following list provides a quick guide for the typical steps involved in setting up a transfer.

1. Initiating a DMA/QDMA channel
   a. Determine the type of channel (QDMA or DMA) to be used.
   b. 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.
   c. 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 11.3.7.1.)
   d. 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.
   e. 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.
2. Parameter set setup
   a. 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.

3. Interrupt setup
   a. Enable the interrupt in the EDMA_TPCC_IER / EDMA_TPCC_IERH by writing into EDMA_TPCC_IESR / EDMA_TPCC_IESRH.
   b. 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.
   c. Set up the interrupt controller properly to receive the expected EDMA interrupt.
4. Initiate transfer
   a. 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.
5. Wait for completion
   a. 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.

b. 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.

11.5 EDMA Debug Checklist and Programming Tips

This section lists some tips to keep in mind while debugging applications using the EDMA controller.

11.5.1 EDMA Debug Checklist

Table 11-16 provides some common issues and their probable causes and resolutions.

Table 11-16. 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 11.3.9.1.2 *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).

11.5.2 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).

11.6 EDMA Event Map

11.6.1 APPSS TPCC_A Event Map

Table 11-17. APPSS TPCC_A Event Map

S No	TPCC0 DMA triggers	Description
0	SPI1_DMA_RX_REQ[0]	SPI1 DMA RX Request (DMAR in McSPI_1 IP channel-0)
1	SPI1_DMA_TX_REQ[0]	SPI1 DMA TX Request (DMAW in McSPI_1 IP channel-0)
2	SPI2_DMA_RX_REQ[0]	SPI2 DMA RX Request (DMAR in McSPI_2 IP channel-0)
3	SPI2_DMA_TX_REQ[0]	SPI2 DMA TX Request (DMAW in McSPI_2 IP channel-0)
4	SCI1_DMA_RX_REQ	SCI1 DMA RX Request
5	SCI1_DMA_TX_REQ	SCI1 DMA TX Request
6	LIN_DMA_RX_REQ	LIN DMA RX Request
7	LIN_DMA_TX_REQ	LIN DMA TX Request

Table 11-17. APPSS TPCC_A Event Map (continued)

8	MCAN_DMA_REQ0	MCAN DMA Request 0
9	MCAN_DMA_REQ1	MCAN DMA Request 1
10	MCAN_FE_INT1	MCAN filter event 1
11	MCAN_FE_INT2	MCAN filter event 2
12	MCAN_FE_INT3	MCAN filter event 3
13	MCAN_FE_INT4	MCAN filter event 4
14	MCAN_FE_INT5	MCAN filter event 5
15	MCAN_FE_INT6	MCAN filter event 6
16	MCAN_FE_INT7	MCAN filter event 7
17	I2C_DMA_REQ0	I2C DMA Request 0
18	I2C_DMA_REQ1	I2C DMA Request 1
19	GIO_INT0	Interrupt trigger from GIO[0]
20	GIO_INT1	Interrupt trigger from GIO[1]
21	APP_RTI1_DMA_REQ0	APP RTI1 DMA Request 0
22	APP_RTI1_DMA_REQ1	APP RTI1 DMA Request 1
23	APP_RTI1_DMA_REQ2	APP RTI1 DMA Request 2
24	APP_RTI1_DMA_REQ3	APP RTI1 DMA Request 3
25	APP_RTI2_DMA_REQ0	APP RTI2 DMA Request 0
26	APP_RTI2_DMA_REQ1	APP RTI2 DMA Request 1
27	APP_RTI2_DMA_REQ2	APP RTI2 DMA Request 2
28	APP_RTI2_DMA_REQ3	APP RTI2 DMA Request 3
29	MCRC_DMA_REQ0	MCRC DMA Request 0
30	MCRC_DMA_REQ1	MCRC DMA Request 1
31	QSPI_DMA_REQ	QSPI DMA Request
32	PWM_DMA_REQ0	PWM DMA Request 0
33	PWM_DMA_REQ1	PWM DMA Request 1
34	SCI2_DMA_RX_REQ	SCI2 DMA RX Request
35	SCI2_DMA_TX_REQ	SCI2 DMA TX Request
36	FRAMETIMER_FRAME_START	Frametimer frame start interrupt from timing engine
37	CHIP_AVAIL_IRQ	Chirp Available Interrupt from ADCBUF Ping/ Pong Buffer
38	CHIRPTIMER_CHIRP_END	Chirptimer chirp end interrupt from timing engine
39	CHIRPTIMER_CHIRP_START	Chirptimer chirp start interrupt from timing engine
40	CHIRPTIMER_FRAME_END	Chirptimer frame end interrupt from timing engine
41	ADC_VALID_START	ADC valid start interrupt from timing engine
42	DTHE_SHA_DMA_REQ0	Reserved
43	DTHE_SHA_DMA_REQ1	Reserved

Table 11-17. APPSS TPCC_A Event Map (continued)

44	DTHE_SHA_DMA_REQ2	Reserved
45	DTHE_SHA_DMA_REQ3	Reserved
46	DTHE_SHA_DMA_REQ4	Reserved
47	DTHE_SHA_DMA_REQ5	Reserved
48	DTHE_AES_DMA_REQ0	Reserved
49	DTHE_AES_DMA_REQ1	Reserved
50	DTHE_AES_DMA_REQ2	Reserved
51	DTHE_AES_DMA_REQ3	Reserved
52	DTHE_AES_DMA_REQ4	Reserved
53	DTHE_AES_DMA_REQ5	Reserved
54	DTHE_AES_DMA_REQ6	Reserved
55	DTHE_AES_DMA_REQ7	Reserved
56	RESERVED	Reserved
57	FEC_INTR[0]	Reserved
58	FEC_INTR[1]	Reserved
59	FEC_INTR[2]	Reserved
60	FEC_INTR[3]	Reserved
61	RESERVED	Reserved
62	SPI1_DMA_RX_REQ[1] OR SPI2_DMA_RX_REQ[1]	The interrupts are logically OR-ed. The DMA request which is not needed should be disabled from the corresponding register - MCSPI_CH(i)CONF (DMAR) in McSPI IP.. Please refer to the IP documentation.
63	SPI1_DMA_TX_REQ[1] OR SPI2_DMA_TX_REQ[1]	The interrupts are logically OR-ed. The DMA request which is not needed should be disabled from the corresponding register - MCSPI_CH(i)CONF in McSPI IP.. Please refer to the IP documentation.

ADVANCE INFORMATION

11.6.2 HWASS TPCC_B Event Map

Table 11-18. HWASS TPCC_B Event Map

S No	TPCC1 (HWASS) DMA Requests	Description
0	FRAMETIMER_FRAME_START	Frametimer frame start interrupt from timing engine
1	CHIP_AVAIL_IRQ	Chirp Available Interrupt from ADCBUF Ping/Pong Buffer
2	CHIRPTIMER_CHIRP_END	Chirptimer chirp end interrupt from timing engine
3	CHIRPTIMER_CHIRP_START	Chirptimer chirp start interrupt from timing engine
4	CHIRPTIMER_FRAME_END	Chirptimer frame end interrupt from timing engine
5	ADC_VALID_START	Adc valid start interrupt from timing engine

Table 11-18. HWASS TPCC_B Event Map (continued)

6	DSS_HW_ACC_CHANNEL_TRIGGER_0	HWA DMA channel 0 interrupt from hwa
7	DSS_HW_ACC_CHANNEL_TRIGGER_1	HWA DMA channel 1 interrupt from hwa
8	DSS_HW_ACC_CHANNEL_TRIGGER_2	HWA DMA channel 2 interrupt from hwa
9	DSS_HW_ACC_CHANNEL_TRIGGER_3	HWA DMA channel 3 interrupt from hwa
10	DSS_HW_ACC_CHANNEL_TRIGGER_4	HWA DMA channel 4 interrupt from hwa
11	DSS_HW_ACC_CHANNEL_TRIGGER_5	HWA DMA channel 5 interrupt from hwa
12	DSS_HW_ACC_CHANNEL_TRIGGER_6	HWA DMA channel 6 interrupt from hwa
13	DSS_HW_ACC_CHANNEL_TRIGGER_7	HWA DMA channel 7 interrupt from hwa
14	DSS_HW_ACC_CHANNEL_TRIGGER_8	HWA DMA channel 8 interrupt from hwa
15	DSS_HW_ACC_CHANNEL_TRIGGER_9	HWA DMA channel 9 interrupt from hwa
16	DSS_HW_ACC_CHANNEL_TRIGGER_10	HWA DMA channel 10 interrupt from hwa
17	DSS_HW_ACC_CHANNEL_TRIGGER_11	HWA DMA channel 11 interrupt from hwa
18	DSS_HW_ACC_CHANNEL_TRIGGER_12	HWA DMA channel 12 interrupt from hwa
19	DSS_HW_ACC_CHANNEL_TRIGGER_13	HWA DMA channel 13 interrupt from hwa
20	DSS_HW_ACC_CHANNEL_TRIGGER_14	HWA DMA channel 14 interrupt from hwa
21	DSS_HW_ACC_CHANNEL_TRIGGER_15	HWA DMA channel 15 interrupt from hwa
22	HWA_LOOP_INT	HWA loop completion interrupt from hwa
23	HWA_PARAMDONE_INT	HWA param done interrupt from hwa
24	SPI1_DMA_RX_REQ	SPI1 DMA RX Request
25	SPI1_DMA_TX_REQ	SPI1 DMA TX Request
26	SPI2_DMA_RX_REQ	SPI2 DMA RX Request
27	SPI2_DMA_TX_REQ	SPI2 DMA TX Request

11.7 EDMA Request Map

For the EDMA request map, refer to Section 11.6.

11.8 EDMA Register Manual

11.8.1 EDMA Registers

11.8.1.1 TPCC Registers

Table 11-19 lists the TPCC registers. All register offset addresses not listed in Table 11-19 should be considered as reserved locations and the register contents should not be modified.

Table 11-19. TPCC Registers

Offset	Acronym	Register Name	Section
0h	PID	PID	Section 11.8.1.1.1
4h	CCCFG	CCCFG	Section 11.8.1.1.2
200h	QCHMAPN	QCHMAPN	Section 11.8.1.1.3
240h	DMAQNUMN	DMAQNUMN	Section 11.8.1.1.4
260h	QDMAQNUM	QDMAQNUM	Section 11.8.1.1.5
280h	QUETCMAP	QUETCMAP	Section 11.8.1.1.6
284h	QUEPRI	QUEPRI	Section 11.8.1.1.7
300h	EMR	EMR	Section 11.8.1.1.8
304h	EMRH	EMRH	Section 11.8.1.1.9
308h	EMCR	EMCR	Section 11.8.1.1.10
30Ch	EMCRH	EMCRH	Section 11.8.1.1.11
310h	QEMR	QEMR	Section 11.8.1.1.12
314h	QEMCR	QEMCR	Section 11.8.1.1.13
318h	CCERR	CCERR	Section 11.8.1.1.14
31Ch	CCERRCLR	CCERRCLR	Section 11.8.1.1.15
320h	EEVAL	EEVAL	Section 11.8.1.1.16
340h	DRAEM	DRAEM	Section 11.8.1.1.17
344h	DRAEHM	DRAEHM	Section 11.8.1.1.18
380h	QRAEN	QRAEN	Section 11.8.1.1.19
400h	QNE0	QNE0	Section 11.8.1.1.20
404h	QNE1	QNE1	Section 11.8.1.1.21
408h	QNE2	QNE2	Section 11.8.1.1.22
40Ch	QNE3	QNE3	Section 11.8.1.1.23
410h	QNE4	QNE4	Section 11.8.1.1.24
414h	QNE5	QNE5	Section 11.8.1.1.25
418h	QNE6	QNE6	Section 11.8.1.1.26
41Ch	QNE7	QNE7	Section 11.8.1.1.27
420h	QNE8	QNE8	Section 11.8.1.1.28
424h	QNE9	QNE9	Section 11.8.1.1.29
428h	QNE10	QNE10	Section 11.8.1.1.30
42Ch	QNE11	QNE11	Section 11.8.1.1.31
430h	QNE12	QNE12	Section 11.8.1.1.32
434h	QNE13	QNE13	Section 11.8.1.1.33
438h	QNE14	QNE14	Section 11.8.1.1.34
43Ch	QNE15	QNE15	Section 11.8.1.1.35
600h	QSTATN	QSTATN	Section 11.8.1.1.36
620h	QWMTHRA	QWMTHRA	Section 11.8.1.1.37
640h	CCSTAT	CCSTAT	Section 11.8.1.1.38
700h	AETCTL	AETCTL	Section 11.8.1.1.39
704h	AETSTAT	AETSTAT	Section 11.8.1.1.40
708h	AETCMD	AETCMD	Section 11.8.1.1.41
1000h	ER	ER	Section 11.8.1.1.42
1004h	ERH	ERH	Section 11.8.1.1.43

Table 11-19. TPCC Registers (continued)

Offset	Acronym	Register Name	Section
1008h	ECR	ECR	Section 11.8.1.1.44
100Ch	ECRH	ECRH	Section 11.8.1.1.45
1010h	ESR	ESR	Section 11.8.1.1.46
1014h	ESRH	ESRH	Section 11.8.1.1.47
1018h	CER	CER	Section 11.8.1.1.48
101Ch	CERH	CERH	Section 11.8.1.1.49
1020h	EER	EER	Section 11.8.1.1.50
1024h	EERH	EERH	Section 11.8.1.1.51
1028h	EECR	EECR	Section 11.8.1.1.52
102Ch	EECRH	EECRH	Section 11.8.1.1.53
1030h	EESR	EESR	Section 11.8.1.1.54
1034h	EESRH	EESRH	Section 11.8.1.1.55
1038h	SER	SER	Section 11.8.1.1.56
103Ch	SERH	SERH	Section 11.8.1.1.57
1040h	SECR	SECR	Section 11.8.1.1.58
1044h	SECRH	SECRH	Section 11.8.1.1.59
1050h	IER	IER	Section 11.8.1.1.60
1054h	IERH	IERH	Section 11.8.1.1.61
1058h	IECR	IECR	Section 11.8.1.1.62
105Ch	IECRH	IECRH	Section 11.8.1.1.63
1060h	IESR	IESR	Section 11.8.1.1.64
1064h	IESRH	IESRH	Section 11.8.1.1.65
1068h	IPR	IPR	Section 11.8.1.1.66
106Ch	IPRH	IPRH	Section 11.8.1.1.67
1070h	ICR	ICR	Section 11.8.1.1.68
1074h	ICRH	ICRH	Section 11.8.1.1.69
1078h	IEVAL	IEVAL	Section 11.8.1.1.70
1080h	QER	QER	Section 11.8.1.1.71
1084h	QEER	QEER	Section 11.8.1.1.72
1088h	QEECR	QEECR	Section 11.8.1.1.73
108Ch	QEESR	QEESR	Section 11.8.1.1.74
1090h	QSER	QSER	Section 11.8.1.1.75
1094h	QSECR	QSECR	Section 11.8.1.1.76
2000h	ER_RN	ER_RN	Section 11.8.1.1.77
2004h	ERH_RN	ERH_RN	Section 11.8.1.1.78
2008h	ECR_RN	ECR_RN	Section 11.8.1.1.79
200Ch	ECRH_RN	ECRH_RN	Section 11.8.1.1.80
2010h	ESR_RN	ESR_RN	Section 11.8.1.1.81
2014h	ESRH_RN	ESRH_RN	Section 11.8.1.1.82
2018h	CER_RN	CER_RN	Section 11.8.1.1.83
201Ch	CERH_RN	CERH_RN	Section 11.8.1.1.84
2020h	EER_RN	EER_RN	Section 11.8.1.1.85
2024h	EERH_RN	EERH_RN	Section 11.8.1.1.86
2028h	EECR_RN	EECR_RN	Section 11.8.1.1.87
202Ch	EECRH_RN	EECRH_RN	Section 11.8.1.1.88
2030h	EESR_RN	EESR_RN	Section 11.8.1.1.89
2034h	EESRH_RN	EESRH_RN	Section 11.8.1.1.90

Table 11-19. TPCC Registers (continued)

Offset	Acronym	Register Name	Section
2038h	SER_RN	SER_RN	Section 11.8.1.1.91
203Ch	SERH_RN	SERH_RN	Section 11.8.1.1.92
2040h	SECR_RN	SECR_RN	Section 11.8.1.1.93
2044h	SECRH_RN	SECRH_RN	Section 11.8.1.1.94
2050h	IER_RN	IER_RN	Section 11.8.1.1.95
2054h	IERH_RN	IERH_RN	Section 11.8.1.1.96
2058h	IECR_RN	IECR_RN	Section 11.8.1.1.97
205Ch	IECRH_RN	IECRH_RN	Section 11.8.1.1.98
2060h	IESR_RN	IESR_RN	Section 11.8.1.1.99
2064h	IESRH_RN	IESRH_RN	Section 11.8.1.1.100
2068h	IPR_RN	IPR_RN	Section 11.8.1.1.101
206Ch	IPRH_RN	IPRH_RN	Section 11.8.1.1.102
2070h	ICR_RN	ICR_RN	Section 11.8.1.1.103
2074h	ICRH_RN	ICRH_RN	Section 11.8.1.1.104
2078h	IEVAL_RN	IEVAL_RN	Section 11.8.1.1.105
2080h	QER_RN	QER_RN	Section 11.8.1.1.106
2084h	QEER_RN	QEER_RN	Section 11.8.1.1.107
2088h	QEECR_RN	QEECR_RN	Section 11.8.1.1.108
208Ch	QEESR_RN	QEESR_RN	Section 11.8.1.1.109
2090h	QSER_RN	QSER_RN	Section 11.8.1.1.110
2094h	QSECR_RN	QSECR_RN	Section 11.8.1.1.111
4000h	OPT	OPT	Section 11.8.1.1.112
4004h	SRC	SRC	Section 11.8.1.1.113
4008h	ABCNT	ABCNT	Section 11.8.1.1.114
400Ch	DST	DST	Section 11.8.1.1.115
4010h	BIDX	BIDX	Section 11.8.1.1.116
4014h	LNK	LNK	Section 11.8.1.1.117
4018h	CIDX	CIDX	Section 11.8.1.1.118
401Ch	CCNT	CCNT	Section 11.8.1.1.119

11.8.1.1.1 PID Register (Offset = 0h) [reset = 4001AB00h]

PID is shown in Figure 11-26 and described in Table 11-20.

Return to the Table 11-19.

Peripheral ID Register

Figure 11-26. 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					

Table 11-20. 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

11.8.1.1.2 CCCFG Register (Offset = 4h) [reset = 00213445h]

CCCFG is shown in Figure 11-27 and described in Table 11-21.

Return to the Table 11-19.

CC Configuration Register

Figure 11-27. CCCFG Register

31	30	29	28	27	26	25	24
RES2						MPEXIST	CHMAPEXIST
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RES3		NUMREGN		RES4	NUMTC		
R-0h		R-2h		R-0h	R-1h		
15	14	13	12	11	10	9	8
RES5	NUMPAENTRY			RES6	NUMINTCH		
R-0h	R-3h			R-0h	R-4h		
7	6	5	4	3	2	1	0
RES7	NUMQDMACH			RES8	NUMDMACH		
R-0h	R-4h			R-0h	R-5h		

Table 11-21. CCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RES2	R	0h	RESERVE FIELD
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	RESERVE FIELD
21-20	NUMREGN	R	2h	Number of MP and Shadow regions
19	RES4	R	0h	RESERVE FIELD
18-16	NUMTC	R	1h	Number of Queues/Number of TCs
15	RES5	R	0h	RESERVE FIELD
14-12	NUMPAENTRY	R	3h	Number of PaRAM entries
11	RES6	R	0h	RESERVE FIELD
10-8	NUMINTCH	R	4h	Number of Interrupt Channels
7	RES7	R	0h	RESERVE FIELD
6-4	NUMQDMACH	R	4h	Number of QDMA Channels
3	RES8	R	0h	RESERVE FIELD
2-0	NUMDMACH	R	5h	Number of DMA Channels

11.8.1.1.3 QCHMAPN Register (Offset = 200h) [reset = 0h]

QCHMAPN is shown in Figure 11-28 and described in Table 11-22.

Return to the Table 11-19.

QDMA Channel N Mapping Register

Figure 11-28. QCHMAPN 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			PAENTRY				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
PAENTRY			TRWORD			RESERVED	
R/W-0h			R/W-0h			R-	

Table 11-22. 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	0h	

11.8.1.1.4 DMAQNUMN Register (Offset = 240h) [reset = 0h]

DMAQNUMN is shown in Figure 11-29 and described in Table 11-23.

Return to the Table 11-19.

DMA Queue Number Register n Contains the Event queue number to be used for the corresponding DMA Channel.

Figure 11-29. 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		

Table 11-23. 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

11.8.1.1.5 QDMAQNUM Register (Offset = 260h) [reset = 0h]

QDMAQNUM is shown in Figure 11-30 and described in Table 11-24.

Return to the Table 11-19.

QDMA Queue Number Register Contains the Event queue number to be used for the corresponding QDMA Channel.

Figure 11-30. 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		

Table 11-24. 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

11.8.1.1.6 QUETCMAP Register (Offset = 280h) [reset = 10h]

QUETCMAP is shown in Figure 11-31 and described in Table 11-25.

Return to the Table 11-19.

Queue to TC Mapping

Figure 11-31. 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		

Table 11-25. 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.

11.8.1.1.7 QUEPRI Register (Offset = 284h) [reset = 0h]

QUEPRI is shown in Figure 11-32 and described in Table 11-26.

Return to the Table 11-19.

Queue Priority

Figure 11-32. 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		

Table 11-26. 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.

11.8.1.1.8 EMR Register (Offset = 300h) [reset = 0h]

EMR is shown in Figure 11-33 and described in Table 11-27.

Return to the Table 11-19.

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 11-33. 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

Table 11-27. 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 11-27. 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

11.8.1.1.9 EMRH Register (Offset = 304h) [reset = 0h]

EMRH is shown in Figure 11-34 and described in Table 11-28.

Return to the Table 11-19.

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 11-34. 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

Table 11-28. 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 11-28. 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

11.8.1.1.10 EMCR Register (Offset = 308h) [reset = 0h]

EMCR is shown in Figure 11-35 and described in Table 11-29.

Return to the Table 11-19.

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 11-35. 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

Table 11-29. 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
4	E4	W	0h	Event Missed Clear #4

Table 11-29. EMCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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

11.8.1.1.11 EMCRH Register (Offset = 30Ch) [reset = 0h]

EMCRH is shown in Figure 11-36 and described in Table 11-30.

Return to the Table 11-19.

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 11-36. 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

Table 11-30. 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 11-30. 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

11.8.1.1.12 QEMR Register (Offset = 310h) [reset = 0h]

QEMR is shown in Figure 11-37 and described in Table 11-31.

Return to the Table 11-19.

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 11-37. 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

Table 11-31. 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

11.8.1.1.13 QEMCR Register (Offset = 314h) [reset = 0h]

QEMCR is shown in Figure 11-38 and described in Table 11-32.

Return to the Table 11-19.

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 11-38. 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

Table 11-32. 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

11.8.1.1.14 CCERR Register (Offset = 318h) [reset = 0h]

CCERR is shown in Figure 11-39 and described in Table 11-33.

Return to the Table 11-19.

CC Error Register

Figure 11-39. 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

Table 11-33. 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 11-33. 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.

11.8.1.1.15 CCERRCLR Register (Offset = 31Ch) [reset = 0h]

CCERRCLR is shown in Figure 11-40 and described in Table 11-34.

Return to the Table 11-19.

CC Error Clear Register

Figure 11-40. 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

Table 11-34. 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.

Table 11-34. CCERRCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

11.8.1.1.16 EEVAL Register (Offset = 320h) [reset = 0h]

EEVAL is shown in Figure 11-41 and described in Table 11-35.

Return to the Table 11-19.

Error Eval Register

Figure 11-41. 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

Table 11-35. 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.

11.8.1.1.17 DRAEM Register (Offset = 340h) [reset = 0h]

DRAEM is shown in Figure 11-42 and described in Table 11-36.

Return to the Table 11-19.

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 11-42. 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

Table 11-36. 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
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

Table 11-36. DRAEM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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

11.8.1.1.18 DRAEHM Register (Offset = 344h) [reset = 0h]

DRAEHM is shown in Figure 11-43 and described in Table 11-37.

Return to the Table 11-19.

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 11-43. 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

Table 11-37. 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
11	E43	R/W	0h	DMA Region Access enable for Region M bit #43

Table 11-37. DRAEHM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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

11.8.1.1.19 QRAEN Register (Offset = 380h) [reset = 0h]

QRAEN is shown in Figure 11-44 and described in Table 11-38.

Return to the Table 11-19.

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 11-44. 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

Table 11-38. 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

11.8.1.1.20 QNE0 Register (Offset = 400h) [reset = 0h]

QNE0 is shown in Figure 11-45 and described in Table 11-39.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 0

Figure 11-45. 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					

Table 11-39. 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).

11.8.1.1.21 QNE1 Register (Offset = 404h) [reset = 0h]

QNE1 is shown in Figure 11-46 and described in Table 11-40.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 1

Figure 11-46. 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					

Table 11-40. 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).

11.8.1.1.22 QNE2 Register (Offset = 408h) [reset = 0h]

QNE2 is shown in Figure 11-47 and described in Table 11-41.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 2

Figure 11-47. 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					

Table 11-41. 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).

11.8.1.1.23 QNE3 Register (Offset = 40Ch) [reset = 0h]

QNE3 is shown in Figure 11-48 and described in Table 11-42.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 3

Figure 11-48. 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					

Table 11-42. 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).

11.8.1.1.24 QNE4 Register (Offset = 410h) [reset = 0h]

QNE4 is shown in Figure 11-49 and described in Table 11-43.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 4

Figure 11-49. 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					

Table 11-43. 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).

11.8.1.1.25 QNE5 Register (Offset = 414h) [reset = 0h]

QNE5 is shown in Figure 11-50 and described in Table 11-44.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 5

Figure 11-50. 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					

Table 11-44. 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).

11.8.1.1.26 QNE6 Register (Offset = 418h) [reset = 0h]

QNE6 is shown in Figure 11-51 and described in Table 11-45.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 6

Figure 11-51. 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					

Table 11-45. 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).

11.8.1.1.27 QNE7 Register (Offset = 41Ch) [reset = 0h]

QNE7 is shown in Figure 11-52 and described in Table 11-46.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 7

Figure 11-52. 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					

Table 11-46. 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).

11.8.1.1.28 QNE8 Register (Offset = 420h) [reset = 0h]

QNE8 is shown in Figure 11-53 and described in Table 11-47.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 8

Figure 11-53. 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					

Table 11-47. 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).

11.8.1.1.29 QNE9 Register (Offset = 424h) [reset = 0h]

QNE9 is shown in Figure 11-54 and described in Table 11-48.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 9

Figure 11-54. 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					

Table 11-48. 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).

11.8.1.1.30 QNE10 Register (Offset = 428h) [reset = 0h]

QNE10 is shown in Figure 11-55 and described in Table 11-49.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 0

Figure 11-55. 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					

Table 11-49. 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).

11.8.1.1.31 QNE11 Register (Offset = 42Ch) [reset = 0h]

QNE11 is shown in Figure 11-56 and described in Table 11-50.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 11

Figure 11-56. 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					

Table 11-50. 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).

11.8.1.1.32 QNE12 Register (Offset = 430h) [reset = 0h]

QNE12 is shown in Figure 11-57 and described in Table 11-51.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 12

Figure 11-57. 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					

Table 11-51. 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).

11.8.1.1.33 QNE13 Register (Offset = 434h) [reset = 0h]

QNE13 is shown in Figure 11-58 and described in Table 11-52.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 13

Figure 11-58. 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					

Table 11-52. 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).

11.8.1.1.34 QNE14 Register (Offset = 438h) [reset = 0h]

QNE14 is shown in Figure 11-59 and described in Table 11-53.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 14

Figure 11-59. 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					

Table 11-53. 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).

11.8.1.1.35 QNE15 Register (Offset = 43Ch) [reset = 0h]

QNE15 is shown in Figure 11-60 and described in Table 11-54.

Return to the Table 11-19.

Event Queue Entry Diagram for Queue n - Entry 15

Figure 11-60. 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					

Table 11-54. 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).

11.8.1.1.36 QSTATN Register (Offset = 600h) [reset = 0h]

QSTATN is shown in Figure 11-61 and described in Table 11-55.

Return to the Table 11-19.

QSTATn Register Set

Figure 11-61. 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			

Table 11-55. 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)

11.8.1.1.37 QWMTHRA Register (Offset = 620h) [reset = 1010h]

QWMTHRA is shown in Figure 11-62 and described in Table 11-56.

Return to the Table 11-19.

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 11-62. 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				

Table 11-56. 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

11.8.1.1.38 CCSTAT Register (Offset = 640h) [reset = 0h]

CCSTAT is shown in Figure 11-63 and described in Table 11-57.

Return to the Table 11-19.

CC Status Register

Figure 11-63. 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

Table 11-57. 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

Table 11-57. CCSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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
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.

11.8.1.1.39 AETCTL Register (Offset = 700h) [reset = 0h]

AETCTL is shown in Figure 11-64 and described in Table 11-58.

Return to the Table 11-19.

Advanced Event Trigger Control

Figure 11-64. 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					

Table 11-58. 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)

11.8.1.1.40 AETSTAT Register (Offset = 704h) [reset = 0h]

AETSTAT is shown in Figure 11-65 and described in Table 11-59.

Return to the Table 11-19.

Advanced Event Trigger Stat

Figure 11-65. 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

Table 11-59. 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.

11.8.1.1.41 AETCMD Register (Offset = 708h) [reset = 0h]

AETCMD is shown in Figure 11-66 and described in Table 11-60.

Return to the Table 11-19.

AET Command

Figure 11-66. 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

Table 11-60. 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..

11.8.1.1.42 ER Register (Offset = 1000h) [reset = 0h]

ER is shown in Figure 11-67 and described in Table 11-61.

Return to the Table 11-19.

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 11-67. 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

Table 11-61. 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 11-61. 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

11.8.1.1.43 ERH Register (Offset = 1004h) [reset = 0h]

ERH is shown in Figure 11-68 and described in Table 11-62.

Return to the Table 11-19.

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 11-68. 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

Table 11-62. 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 11-62. 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

11.8.1.1.44 ECR Register (Offset = 1008h) [reset = 0h]

ECR is shown in Figure 11-69 and described in Table 11-63.

Return to the Table 11-19.

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 11-69. 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

Table 11-63. 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 11-63. 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

11.8.1.1.45 ECRH Register (Offset = 100Ch) [reset = 0h]

ECRH is shown in Figure 11-70 and described in Table 11-64.

Return to the Table 11-19.

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 11-70. 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

Table 11-64. 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 11-64. 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

11.8.1.1.46 ESR Register (Offset = 1010h) [reset = 0h]

ESR is shown in Figure 11-71 and described in Table 11-65.

Return to the Table 11-19.

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 11-71. 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

Table 11-65. 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 11-65. 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

11.8.1.1.47 ESRH Register (Offset = 1014h) [reset = 0h]

ESRH is shown in Figure 11-72 and described in Table 11-66.

Return to the Table 11-19.

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 11-72. 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

Table 11-66. 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 11-66. 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

11.8.1.1.48 CER Register (Offset = 1018h) [reset = 0h]

CER is shown in Figure 11-73 and described in Table 11-67.

Return to the Table 11-19.

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 11-73. 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

Table 11-67. 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
7	E7	R	0h	Event #7

Table 11-67. CER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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

11.8.1.1.49 CERH Register (Offset = 101Ch) [reset = 0h]

CERH is shown in Figure 11-74 and described in Table 11-68.

Return to the Table 11-19.

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 11-74. 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

Table 11-68. 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 11-68. 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

11.8.1.1.50 EER Register (Offset = 1020h) [reset = 0h]

EER is shown in Figure 11-75 and described in Table 11-69.

Return to the Table 11-19.

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 11-75. 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

Table 11-69. 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 11-69. 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

11.8.1.1.51 EERH Register (Offset = 1024h) [reset = 0h]

EERH is shown in Figure 11-76 and described in Table 11-70.

Return to the Table 11-19.

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 11-76. 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

Table 11-70. 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 11-70. 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

11.8.1.1.52 EECR Register (Offset = 1028h) [reset = 0h]

EECR is shown in Figure 11-77 and described in Table 11-71.

Return to the Table 11-19.

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 11-77. 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

Table 11-71. 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 11-71. 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

11.8.1.1.53 EECRH Register (Offset = 102Ch) [reset = 0h]

EECRH is shown in Figure 11-78 and described in Table 11-72.

Return to the Table 11-19.

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 11-78. 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

Table 11-72. 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 11-72. 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

11.8.1.1.54 EESR Register (Offset = 1030h) [reset = 0h]

EESR is shown in Figure 11-79 and described in Table 11-73.

Return to the Table 11-19.

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 11-79. 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

Table 11-73. 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 11-73. 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

11.8.1.1.55 EESRH Register (Offset = 1034h) [reset = 0h]

EESRH is shown in Figure 11-80 and described in Table 11-74.

Return to the Table 11-19.

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 11-80. 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

Table 11-74. 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 11-74. 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

11.8.1.1.56 SER Register (Offset = 1038h) [reset = 0h]

SER is shown in Figure 11-81 and described in Table 11-75.

Return to the Table 11-19.

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 11-81. 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

Table 11-75. 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 11-75. 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

11.8.1.1.57 SERH Register (Offset = 103Ch) [reset = 0h]

SERH is shown in Figure 11-82 and described in Table 11-76.

Return to the Table 11-19.

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 11-82. 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

Table 11-76. 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 11-76. 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

11.8.1.1.58 SECR Register (Offset = 1040h) [reset = 0h]

SECR is shown in Figure 11-83 and described in Table 11-77.

Return to the Table 11-19.

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 11-83. 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

Table 11-77. 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 11-77. 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

11.8.1.1.59 SECRH Register (Offset = 1044h) [reset = 0h]

SECRH is shown in Figure 11-84 and described in Table 11-78.

Return to the Table 11-19.

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 11-84. 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

Table 11-78. 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
4	E36	W	0h	Event #36

Table 11-78. SECRH 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

11.8.1.1.60 IER Register (Offset = 1050h) [reset = 0h]

IER is shown in Figure 11-85 and described in Table 11-79.

Return to the Table 11-19.

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 11-85. 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

Table 11-79. 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 11-79. 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

11.8.1.1.61 IERH Register (Offset = 1054h) [reset = 0h]

IERH is shown in Figure 11-86 and described in Table 11-80.

Return to the Table 11-19.

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 11-86. 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

Table 11-80. 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 11-80. 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

11.8.1.1.62 IECR Register (Offset = 1058h) [reset = 0h]

IECR is shown in Figure 11-87 and described in Table 11-81.

Return to the Table 11-19.

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 11-87. 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

Table 11-81. 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 11-81. 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

11.8.1.1.63 IECRH Register (Offset = 105Ch) [reset = 0h]

IECRH is shown in Figure 11-88 and described in Table 11-82.

Return to the Table 11-19.

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 11-88. 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

Table 11-82. 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 11-82. 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

11.8.1.1.64 IESR Register (Offset = 1060h) [reset = 0h]

IESR is shown in Figure 11-89 and described in Table 11-83.

Return to the Table 11-19.

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 11-89. 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

Table 11-83. 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 11-83. 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

11.8.1.1.65 IESRH Register (Offset = 1064h) [reset = 0h]

IESRH is shown in Figure 11-90 and described in Table 11-84.

Return to the Table 11-19.

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 11-90. 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

Table 11-84. 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 11-84. 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

11.8.1.1.66 IPR Register (Offset = 1068h) [reset = 0h]

IPR is shown in Figure 11-91 and described in Table 11-85.

Return to the Table 11-19.

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 11-91. 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

Table 11-85. 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 11-85. 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

11.8.1.1.67 IPRH Register (Offset = 106Ch) [reset = 0h]

IPRH is shown in Figure 11-92 and described in Table 11-86.

Return to the Table 11-19.

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 11-92. 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

Table 11-86. 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 11-86. 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

11.8.1.1.68 ICR Register (Offset = 1070h) [reset = 0h]

ICR is shown in Figure 11-93 and described in Table 11-87.

Return to the Table 11-19.

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 11-93. 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

Table 11-87. 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 11-87. 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

11.8.1.1.69 ICRH Register (Offset = 1074h) [reset = 0h]

ICRH is shown in Figure 11-94 and described in Table 11-88.

Return to the Table 11-19.

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 11-94. 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

Table 11-88. 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
4	I36	W	0h	Interrupt associated with TCC #36

Table 11-88. ICRH 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

11.8.1.1.70 IEVAL Register (Offset = 1078h) [reset = 0h]

IEVAL is shown in Figure 11-95 and described in Table 11-89.

Return to the Table 11-19.

Interrupt Eval Register

Figure 11-95. 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

Table 11-89. 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..

11.8.1.1.71 QER Register (Offset = 1080h) [reset = 0h]

QER is shown in Figure 11-96 and described in Table 11-90.

Return to the Table 11-19.

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 11-96. 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

Table 11-90. 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

11.8.1.1.72 QEER Register (Offset = 1084h) [reset = 0h]

QEER is shown in Figure 11-97 and described in Table 11-91.

Return to the Table 11-19.

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 11-97. 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

Table 11-91. 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

11.8.1.1.73 QEECR Register (Offset = 1088h) [reset = 0h]

QEECR is shown in Figure 11-98 and described in Table 11-92.

Return to the Table 11-19.

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 11-98. 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

Table 11-92. 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

11.8.1.1.74 QEESR Register (Offset = 108Ch) [reset = 0h]

QEESR is shown in Figure 11-99 and described in Table 11-93.

Return to the Table 11-19.

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 11-99. 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

Table 11-93. 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

11.8.1.1.75 QSER Register (Offset = 1090h) [reset = 0h]

QSER is shown in Figure 11-100 and described in Table 11-94.

Return to the Table 11-19.

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 11-100. 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

Table 11-94. 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

11.8.1.1.76 QSECR Register (Offset = 1094h) [reset = 0h]

QSECR is shown in Figure 11-101 and described in Table 11-95.

Return to the Table 11-19.

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 11-101. 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

Table 11-95. 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

11.8.1.1.77 ER_RN Register (Offset = 2000h) [reset = 0h]

ER_RN is shown in Figure 11-102 and described in Table 11-96.

Return to the Table 11-19.

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 11-102. 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

Table 11-96. 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 11-96. 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

11.8.1.1.78 ERH_RN Register (Offset = 2004h) [reset = 0h]

ERH_RN is shown in Figure 11-103 and described in Table 11-97.

Return to the Table 11-19.

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 11-103. 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

Table 11-97. 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 11-97. 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

11.8.1.1.79 ECR_RN Register (Offset = 2008h) [reset = 0h]

ECR_RN is shown in Figure 11-104 and described in Table 11-98.

Return to the Table 11-19.

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 11-104. 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

Table 11-98. 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 11-98. 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

11.8.1.1.80 ECRH_RN Register (Offset = 200Ch) [reset = 0h]

ECRH_RN is shown in Figure 11-105 and described in Table 11-99.

Return to the Table 11-19.

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 11-105. 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

Table 11-99. 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 11-99. 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

11.8.1.1.81 ESR_RN Register (Offset = 2010h) [reset = 0h]

ESR_RN is shown in Figure 11-106 and described in Table 11-100.

Return to the Table 11-19.

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 11-106. 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

Table 11-100. 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 11-100. 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

11.8.1.1.82 ESRH_RN Register (Offset = 2014h) [reset = 0h]

ESRH_RN is shown in Figure 11-107 and described in Table 11-101.

Return to the Table 11-19.

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 11-107. 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

Table 11-101. 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 11-101. 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

11.8.1.1.83 CER_RN Register (Offset = 2018h) [reset = 0h]

CER_RN is shown in Figure 11-108 and described in Table 11-102.

Return to the Table 11-19.

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 11-108. 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

Table 11-102. 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
7	E7	R	0h	Event #7

Table 11-102. CER_RN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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

11.8.1.1.84 CERH_RN Register (Offset = 201Ch) [reset = 0h]

CERH_RN is shown in Figure 11-109 and described in Table 11-103.

Return to the Table 11-19.

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 11-109. 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

Table 11-103. 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 11-103. 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

11.8.1.1.85 EER_RN Register (Offset = 2020h) [reset = 0h]

EER_RN is shown in Figure 11-110 and described in Table 11-104.

Return to the Table 11-19.

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 11-110. 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

Table 11-104. 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 11-104. 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

11.8.1.1.86 EERH_RN Register (Offset = 2024h) [reset = 0h]

EERH_RN is shown in Figure 11-111 and described in Table 11-105.

Return to the Table 11-19.

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 11-111. 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

Table 11-105. 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 11-105. 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

11.8.1.1.87 EECR_RN Register (Offset = 2028h) [reset = 0h]

EECR_RN is shown in Figure 11-112 and described in Table 11-106.

Return to the Table 11-19.

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 11-112. 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

Table 11-106. 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 11-106. 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

11.8.1.1.88 EECRH_RN Register (Offset = 202Ch) [reset = 0h]

EECRH_RN is shown in Figure 11-113 and described in Table 11-107.

Return to the Table 11-19.

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 11-113. 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

Table 11-107. 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 11-107. 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

11.8.1.1.89 EESR_RN Register (Offset = 2030h) [reset = 0h]

EESR_RN is shown in Figure 11-114 and described in Table 11-108.

Return to the Table 11-19.

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 11-114. 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

Table 11-108. 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 11-108. 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

11.8.1.1.90 EESRH_RN Register (Offset = 2034h) [reset = 0h]

EESRH_RN is shown in Figure 11-115 and described in Table 11-109.

Return to the Table 11-19.

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 11-115. 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

Table 11-109. 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 11-109. 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

11.8.1.1.91 SER_RN Register (Offset = 2038h) [reset = 0h]

SER_RN is shown in Figure 11-116 and described in Table 11-110.

Return to the Table 11-19.

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 11-116. 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

Table 11-110. 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 11-110. 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

11.8.1.1.92 SERH_RN Register (Offset = 203Ch) [reset = 0h]

SERH_RN is shown in Figure 11-117 and described in Table 11-111.

Return to the Table 11-19.

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 11-117. 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

Table 11-111. 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 11-111. 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

11.8.1.1.93 SECR_RN Register (Offset = 2040h) [reset = 0h]

SECR_RN is shown in Figure 11-118 and described in Table 11-112.

Return to the Table 11-19.

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 11-118. 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

Table 11-112. 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 11-112. 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

11.8.1.1.94 SECRH_RN Register (Offset = 2044h) [reset = 0h]

SECRH_RN is shown in Figure 11-119 and described in Table 11-113.

Return to the Table 11-19.

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 11-119. 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

Table 11-113. 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
4	E36	W	0h	Event #36

Table 11-113. SECRH_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

11.8.1.1.95 IER_RN Register (Offset = 2050h) [reset = 0h]

IER_RN is shown in Figure 11-120 and described in Table 11-114.

Return to the Table 11-19.

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 11-120. 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

Table 11-114. 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 11-114. 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

11.8.1.1.96 IERH_RN Register (Offset = 2054h) [reset = 0h]

IERH_RN is shown in Figure 11-121 and described in Table 11-115.

Return to the Table 11-19.

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 11-121. 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

Table 11-115. 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 11-115. 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

11.8.1.1.97 IECR_RN Register (Offset = 2058h) [reset = 0h]

IECR_RN is shown in Figure 11-122 and described in Table 11-116.

Return to the Table 11-19.

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 11-122. 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

Table 11-116. 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 11-116. 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

11.8.1.1.98 IECRH_RN Register (Offset = 205Ch) [reset = 0h]

IECRH_RN is shown in Figure 11-123 and described in Table 11-117.

Return to the Table 11-19.

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 11-123. 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

Table 11-117. 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 11-117. 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

11.8.1.1.99 IESR_RN Register (Offset = 2060h) [reset = 0h]

IESR_RN is shown in Figure 11-124 and described in Table 11-118.

Return to the Table 11-19.

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 11-124. 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

Table 11-118. 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 11-118. 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

11.8.1.1.100 IESRH_RN Register (Offset = 2064h) [reset = 0h]

IESRH_RN is shown in Figure 11-125 and described in Table 11-119.

Return to the Table 11-19.

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 11-125. 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

Table 11-119. 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 11-119. 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

11.8.1.1.101 IPR_RN Register (Offset = 2068h) [reset = 0h]

IPR_RN is shown in Figure 11-126 and described in Table 11-120.

Return to the Table 11-19.

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 11-126. 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

Table 11-120. 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 11-120. 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

11.8.1.1.102 IPRH_RN Register (Offset = 206Ch) [reset = 0h]

IPRH_RN is shown in Figure 11-127 and described in Table 11-121.

Return to the Table 11-19.

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 11-127. 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

Table 11-121. 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 11-121. 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

11.8.1.1.103 ICR_RN Register (Offset = 2070h) [reset = 0h]

ICR_RN is shown in Figure 11-128 and described in Table 11-122.

Return to the Table 11-19.

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 11-128. 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

Table 11-122. 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 11-122. 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

11.8.1.1.104 ICRH_RN Register (Offset = 2074h) [reset = 0h]

ICRH_RN is shown in Figure 11-129 and described in Table 11-123.

Return to the Table 11-19.

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 11-129. 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

Table 11-123. 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
4	I36	W	0h	Interrupt associated with TCC #36

Table 11-123. ICRH_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

11.8.1.1.105 IEVAL_RN Register (Offset = 2078h) [reset = 0h]

IEVAL_RN is shown in Figure 11-130 and described in Table 11-124.

Return to the Table 11-19.

Interrupt Eval Register

Figure 11-130. 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

Table 11-124. 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..

11.8.1.1.106 QER_RN Register (Offset = 2080h) [reset = 0h]

QER_RN is shown in Figure 11-131 and described in Table 11-125.

Return to the Table 11-19.

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 11-131. 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

Table 11-125. 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

11.8.1.1.107 QEER_RN Register (Offset = 2084h) [reset = 0h]

QEER_RN is shown in Figure 11-132 and described in Table 11-126.

Return to the Table 11-19.

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 11-132. 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

Table 11-126. 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

11.8.1.1.108 QEECR_RN Register (Offset = 2088h) [reset = 0h]

QEECR_RN is shown in Figure 11-133 and described in Table 11-127.

Return to the Table 11-19.

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 11-133. 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

Table 11-127. 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

11.8.1.1.109 QEESR_RN Register (Offset = 208Ch) [reset = 0h]

QEESR_RN is shown in Figure 11-134 and described in Table 11-128.

Return to the Table 11-19.

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 11-134. 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

Table 11-128. 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

11.8.1.1.110 QSER_RN Register (Offset = 2090h) [reset = 0h]

QSER_RN is shown in Figure 11-135 and described in Table 11-129.

Return to the Table 11-19.

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 11-135. 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

Table 11-129. 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

11.8.1.1.111 QSECR_RN Register (Offset = 2094h) [reset = 0h]

QSECR_RN is shown in Figure 11-136 and described in Table 11-130.

Return to the Table 11-19.

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 11-136. 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

Table 11-130. 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

11.8.1.1.112 OPT Register (Offset = 4000h) [reset = 0h]

OPT is shown in Figure 11-137 and described in Table 11-131.

Return to the Table 11-19.

Options Parameter

Figure 11-137. 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

Table 11-131. 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

Table 11-131. OPT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.
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.

11.8.1.1.113 SRC Register (Offset = 4004h) [reset = 0h]

SRC is shown in Figure 11-138 and described in Table 11-132.

Return to the Table 11-19.

Source Address

Figure 11-138. 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																															

Table 11-132. 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.

11.8.1.1.114 ABCNT Register (Offset = 4008h) [reset = 0h]

ABCNT is shown in Figure 11-139 and described in Table 11-133.

Return to the Table 11-19.

A and B byte count

Figure 11-139. 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															

Table 11-133. 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.

11.8.1.1.115 DST Register (Offset = 400Ch) [reset = 0h]

DST is shown in Figure 11-140 and described in Table 11-134.

Return to the Table 11-19.

Destination Address

Figure 11-140. 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																															

Table 11-134. 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.

11.8.1.1.116 BIDX Register (Offset = 4010h) [reset = 0h]

BIDX is shown in Figure 11-141 and described in Table 11-135.

Return to the Table 11-19.

Register description is not available

Figure 11-141. 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															

Table 11-135. 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.

11.8.1.1.117 LNK Register (Offset = 4014h) [reset = 0h]

LNK is shown in Figure 11-142 and described in Table 11-136.

Return to the Table 11-19.

Link and Reload parameters

Figure 11-142. 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															

Table 11-136. 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.

11.8.1.1.118 CIDX Register (Offset = 4018h) [reset = 0h]

CIDX is shown in Figure 11-143 and described in Table 11-137.

Return to the Table 11-19.

Register description is not available

Figure 11-143. 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															

Table 11-137. 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.

11.8.1.1.119 CCNT Register (Offset = 401Ch) [reset = 0h]

CCNT is shown in Figure 11-144 and described in Table 11-138.

Return to the Table 11-19.

C byte count

Figure 11-144. 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															

Table 11-138. 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.

11.8.1.2 TPTC Registers

Table 11-139 lists the TPTC registers. All register offset addresses not listed in Table 11-139 should be considered as reserved locations and the register contents should not be modified.

Table 11-139. TPTC Registers

Offset	Acronym	Register Name	Section
0h	PID	Peripheral ID Register	Section 11.8.1.2.1
4h	TCCFG	TC Configuration Register	Section 11.8.1.2.2
100h	TCSTAT	TC Status Register	Section 11.8.1.2.3
104h	INTSTAT	Interrupt Status Register	Section 11.8.1.2.4
108h	INTEN	Interrupt Enable Register	Section 11.8.1.2.5
10Ch	INTCLR	Interrupt Clear Register	Section 11.8.1.2.6
110h	INTCMD	Interrupt Command Register	Section 11.8.1.2.7
120h	ERRSTAT	Error Status Register	Section 11.8.1.2.8
124h	ERREN	Error Enable Register	Section 11.8.1.2.9
128h	ERRCLR	Error Clear Register	Section 11.8.1.2.10
12Ch	ERRDET	Error Details Register	Section 11.8.1.2.11
130h	ERRCMD	Error Command Register	Section 11.8.1.2.12
140h	RDRATE	Read Rate Register	Section 11.8.1.2.13
200h	POPT	Prog Set Options	Section 11.8.1.2.14
204h	PSRC	Prog Set Src Address	Section 11.8.1.2.15
208h	PCNT	Prog Set Count	Section 11.8.1.2.16
20Ch	PDST	Prog Set Dst Address	Section 11.8.1.2.17
210h	PBIDX	Prog Set B-Dim Idx	Section 11.8.1.2.18
214h	PMPPRXY	Prog Set Mem Protect Proxy	Section 11.8.1.2.19
240h	SAOPT	Src Actv Set Options	Section 11.8.1.2.20
244h	SASRC	Src Actv Set Src Address	Section 11.8.1.2.21
248h	SACNT	Src Actv Set A-Count	Section 11.8.1.2.22
24Ch	SADST	Src Actv Set Dst Address	Section 11.8.1.2.23
250h	SABIDX	Src Actv Set B-Dim Idx	Section 11.8.1.2.24
254h	SAMPPRXY	Src Actv Set Mem Protect Proxy	Section 11.8.1.2.25
258h	SACNTRLD	Src Actv Set Cnt Reload	Section 11.8.1.2.26
25Ch	SASRCBREF	Src Actv Set Src Addr B-Reference	Section 11.8.1.2.27
260h	SADSTBREF	Src Actv Set Dst Addr B-Reference	Section 11.8.1.2.28
264h	SABCNT	Src Actv Set B-Count	Section 11.8.1.2.29
280h	DFCNTRLD	Dst FIFO Set Cnt Reload	Section 11.8.1.2.30
284h	DFSRCBREF	Dst FIFO Set Src Addr B-Reference	Section 11.8.1.2.31
300h	DFOPT0	Dst FIFO Set Options	Section 11.8.1.2.32
304h	DFSRC0	Dst FIFO Set Src Address	Section 11.8.1.2.33
308h	DFACNT0	Dst FIFO Set A-Count	Section 11.8.1.2.34
30Ch	DFDST0	Dst FIFO Set Dst Address	Section 11.8.1.2.35
310h	DFBIDX0	Dst FIFO Set B-Dim Idx	Section 11.8.1.2.36
314h	DFMPPRXY0	Dst FIFO Set Mem Protect Proxy	Section 11.8.1.2.37
318h	DFBCNT0	Dst FIFO Set B-Count	Section 11.8.1.2.38
340h	DFOPT1	Dst FIFO Set Options	Section 11.8.1.2.39
344h	DFSRC1	Dst FIFO Set Src Address	Section 11.8.1.2.40
348h	DFACNT1	Dst FIFO Set A-Count	Section 11.8.1.2.41
34Ch	DFDST1	Dst FIFO Set Dst Address	Section 11.8.1.2.42
350h	DFBIDX1	Dst FIFO Set B-Dim Idx	Section 11.8.1.2.43

Table 11-139. TPTC Registers (continued)

Offset	Acronym	Register Name	Section
354h	DFMPPRXY1	Dst FIFO Set Mem Protect Proxy	Section 11.8.1.2.44
358h	DFBCNT1	Dst FIFO Set B-Count	Section 11.8.1.2.45

11.8.1.2.1 PID Register (Offset = 0h) [reset = X]

PID is shown in Figure 11-145 and described in Table 11-140.

Return to the Table 11-139.

Peripheral ID Register

Figure 11-145. PID Register

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

Table 11-140. 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	RESERVED	R	X	
27-16	FUNC	R	0h	Function indicates a software compatible module family.
15-11	RTL	R	1h	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	1h	Minor Revision

11.8.1.2.2 TCCFG Register (Offset = 4h) [reset = X]

TCCFG is shown in Figure 11-146 and described in Table 11-141.

Return to the Table 11-139.

TC Configuration Register

Figure 11-146. TCCFG Register

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

Table 11-141. TCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9-8	DREGDEPTH	R	2h	Dst Register FIFO Depth Parameterization
7-6	RESERVED	R	X	
5-4	BUSWIDTH	R	2h	Bus Width Parameterization
3	RESERVED	R	X	
2-0	FIFOSIZE	R	4h	Fifo Size Parameterization

11.8.1.2.3 TCSTAT Register (Offset = 100h) [reset = X]

TCSTAT is shown in Figure 11-147 and described in Table 11-142.

Return to the Table 11-139.

TC Status Register

Figure 11-147. TCSTAT Register

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

Table 11-142. TCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	X	
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	RESERVED	R	X	
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	RESERVED	R	X	
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	RESERVED	R	X	
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.

Table 11-142. TCSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

11.8.1.2.4 INTSTAT Register (Offset = 104h) [reset = X]

INTSTAT is shown in Figure 11-148 and described in Table 11-143.

Return to the Table 11-139.

Interrupt Status Register

Figure 11-148. INTSTAT Register

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

Table 11-143. INTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	X	
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.

11.8.1.2.5 INTEN Register (Offset = 108h) [reset = X]

INTEN is shown in Figure 11-149 and described in Table 11-144.

Return to the Table 11-139.

Interrupt Enable Register

Figure 11-149. INTEN Register

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

Table 11-144. INTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
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

11.8.1.2.6 INTCLR Register (Offset = 10Ch) [reset = X]

INTCLR is shown in Figure 11-150 and described in Table 11-145.

Return to the Table 11-139.

Interrupt Clear Register

Figure 11-150. INTCLR Register

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

Table 11-145. INTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	X	
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

11.8.1.2.7 INTCMD Register (Offset = 110h) [reset = X]

INTCMD is shown in Figure 11-151 and described in Table 11-146.

Return to the Table 11-139.

Interrupt Command Register

Figure 11-151. INTCMD Register

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

Table 11-146. INTCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	X	
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.

11.8.1.2.8 ERRSTAT Register (Offset = 120h) [reset = X]

ERRSTAT is shown in Figure 11-152 and described in Table 11-147.

Return to the Table 11-139.

Error Status Register

Figure 11-152. ERRSTAT Register

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

Table 11-147. ERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
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	RESERVED	R	X	
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].

11.8.1.2.9 ERREN Register (Offset = 124h) [reset = X]

ERREN is shown in Figure 11-153 and described in Table 11-148.

Return to the Table 11-139.

Error Enable Register

Figure 11-153. ERREN Register

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

Table 11-148. ERREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
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	RESERVED	R/W	X	
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.

11.8.1.2.10 ERRCLR Register (Offset = 128h) [reset = X]

ERRCLR is shown in Figure 11-154 and described in Table 11-149.

Return to the Table 11-139.

Error Clear Register

Figure 11-154. ERRCLR Register

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

Table 11-149. ERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	X	
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	RESERVED	W	X	
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.

11.8.1.2.11 ERRDET Register (Offset = 12Ch) [reset = X]

ERRDET is shown in Figure 11-155 and described in Table 11-150.

Return to the Table 11-139.

Error Details Register

Figure 11-155. ERRDET Register

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

Table 11-150. ERRDET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	X	
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	RESERVED	R	X	
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	RESERVED	R	X	
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.

11.8.1.2.12 ERRCMD Register (Offset = 130h) [reset = X]

ERRCMD is shown in Figure 11-156 and described in Table 11-151.

Return to the Table 11-139.

Error Command Register

Figure 11-156. ERRCMD Register

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

Table 11-151. ERRCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	X	
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.

11.8.1.2.13 RDRATE Register (Offset = 140h) [reset = X]

RDRATE is shown in Figure 11-157 and described in Table 11-152.

Return to the Table 11-139.

Read Rate Register

Figure 11-157. RDRATE Register

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

Table 11-152. RDRATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
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.

11.8.1.2.14 POPT Register (Offset = 200h) [reset = X]

POPT is shown in Figure 11-158 and described in Table 11-153.

Return to the Table 11-139.

Prog Set Options

Figure 11-158. POPT Register

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

Table 11-153. POPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tptc_r_dbg_channel_id) and write (tptc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
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	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

Table 11-153. POPT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

11.8.1.2.15 PSRC Register (Offset = 204h) [reset = 0h]

PSRC is shown in Figure 11-159 and described in Table 11-154.

Return to the Table 11-139.

Prog Set Src Address

Figure 11-159. 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																															

Table 11-154. PSRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDR	R/W	0h	Source address for Program Register Set

11.8.1.2.16 PCNT Register (Offset = 208h) [reset = 0h]

PCNT is shown in Figure 11-160 and described in Table 11-155.

Return to the Table 11-139.

Prog Set Count

Figure 11-160. 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															

Table 11-155. 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.

11.8.1.2.17 PDST Register (Offset = 20Ch) [reset = 0h]

PDST is shown in Figure 11-161 and described in Table 11-156.

Return to the Table 11-139.

Prog Set Dst Address

Figure 11-161. 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																															

Table 11-156. PDST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R/W	0h	Destination address for Program Register Set

11.8.1.2.18 PBIDX Register (Offset = 210h) [reset = 0h]

PBIDX is shown in Figure 11-162 and described in Table 11-157.

Return to the Table 11-139.

Prog Set B-Dim Idx

Figure 11-162. 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															

Table 11-157. 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.

11.8.1.2.19 PMPPRXY Register (Offset = 214h) [reset = X]

PMPPRXY is shown in Figure 11-163 and described in Table 11-158.

Return to the Table 11-139.

Prog Set Mem Protect Proxy

Figure 11-163. PMPPRXY Register

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

Table 11-158. PMPPRXY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
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	RESERVED	R	X	
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.

11.8.1.2.20 SAOPT Register (Offset = 240h) [reset = X]

SAOPT is shown in Figure 11-164 and described in Table 11-159.

Return to the Table 11-139.

Src Actv Set Options

Figure 11-164. SAOPT Register

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

Table 11-159. SAOPT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tptc_r_dbg_channel_id) and write (tptc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
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	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

Table 11-159. SAOPT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

11.8.1.2.21 SASRC Register (Offset = 244h) [reset = 0h]

SASRC is shown in Figure 11-165 and described in Table 11-160.

Return to the Table 11-139.

Src Actv Set Src Address

Figure 11-165. 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																															

Table 11-160. SASRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDR	R	0h	Source address for Source Active Register Set

11.8.1.2.22 SACNT Register (Offset = 248h) [reset = X]

SACNT is shown in Figure 11-166 and described in Table 11-161.

Return to the Table 11-139.

Src Actv Set A-Count

Figure 11-166. 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
RESERVED									ACNT																						
R-X									R-0h																						

Table 11-161. SACNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-0	ACNT	R	0h	A-Dimension count. Number of bytes to be transferred in first dimension.

11.8.1.2.23 SADST Register (Offset = 24Ch) [reset = 0h]

SADST is shown in Figure 11-167 and described in Table 11-162.

Return to the Table 11-139.

Src Actv Set Dst Address

Figure 11-167. SADST Register

31	30	29	28	27	26	25	24	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																															

Table 11-162. SADST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	Destination address for Source Active Register Set

11.8.1.2.24 SABIDX Register (Offset = 250h) [reset = 0h]

SABIDX is shown in Figure 11-168 and described in Table 11-163.

Return to the Table 11-139.

Src Actv Set B-Dim Idx

Figure 11-168. 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															

Table 11-163. SABIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	Dest B-Idx for Source Active 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	0h	Source B-Idx for Source Active 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.

11.8.1.2.25 SAMPPRXY Register (Offset = 254h) [reset = X]

SAMPPRXY is shown in Figure 11-169 and described in Table 11-164.

Return to the Table 11-139.

Src Actv Set Mem Protect Proxy

Figure 11-169. SAMPPRXY Register

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

Table 11-164. SAMPPRXY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
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	RESERVED	R	X	
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.

11.8.1.2.26 SACNTRLD Register (Offset = 258h) [reset = X]

SACNTRLD is shown in Figure 11-170 and described in Table 11-165.

Return to the Table 11-139.

Src Actv Set Cnt Reload

Figure 11-170. 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
RESERVED																ACNTRLD															
R-X																R-0h															

Table 11-165. SACNTRLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
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]

11.8.1.2.27 SASRCBREF Register (Offset = 25Ch) [reset = 0h]

SASRCBREF is shown in Figure 11-171 and described in Table 11-166.

Return to the Table 11-139.

Src Actv Set Src Addr B-Reference

Figure 11-171. 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																															

Table 11-166. 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.

11.8.1.2.28 SADSTBREF Register (Offset = 260h) [reset = 0h]

SADSTBREF is shown in Figure 11-172 and described in Table 11-167.

Return to the Table 11-139.

Src Actv Set Dst Addr B-Reference

Figure 11-172. 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																															

Table 11-167. 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.

11.8.1.2.29 SABCNT Register (Offset = 264h) [reset = X]

SABCNT is shown in Figure 11-173 and described in Table 11-168.

Return to the Table 11-139.

Src Actv Set B-Count

Figure 11-173. SABCNT Register

31	30	29	28	27	26	25	24	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																BCNT															
R-X																R-0h															

Table 11-168. SABCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	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.

11.8.1.2.30 DFCNTRLD Register (Offset = 280h) [reset = X]

DFCNTRLD is shown in Figure 11-174 and described in Table 11-169.

Return to the Table 11-139.

Dst FIFO Set Cnt Reload

Figure 11-174. 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
RESERVED																ACNTRLD															
R-X																R-0h															

Table 11-169. DFCNTRLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
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]

11.8.1.2.31 DFSRCBREF Register (Offset = 284h) [reset = 0h]

DFSRCBREF is shown in Figure 11-175 and described in Table 11-170.

Return to the Table 11-139.

Dst FIFO Set Src Addr B-Reference

Figure 11-175. 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																															

Table 11-170. DFSRCBREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SADDRBREF	R	0h	Source address reference for Destination FIFO 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.

11.8.1.2.32 DFOPT0 Register (Offset = 300h) [reset = X]

DFOPT0 is shown in Figure 11-176 and described in Table 11-171.

Return to the Table 11-139.

Dst FIFO Set Options

Figure 11-176. DFOPT0 Register

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

Table 11-171. DFOPT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tptc_r_dbg_channel_id) and write (tptc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
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	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

Table 11-171. DFOPT0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

11.8.1.2.33 DFSRC0 Register (Offset = 304h) [reset = 0h]

DFSRC0 is shown in Figure 11-177 and described in Table 11-172.

Return to the Table 11-139.

Dst FIFO Set Src Address

Figure 11-177. DFSRC0 Register

31	30	29	28	27	26	25	24	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																															

Table 11-172. DFSRC0 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.

11.8.1.2.34 DFACNT0 Register (Offset = 308h) [reset = X]

DFACNT0 is shown in Figure 11-178 and described in Table 11-173.

Return to the Table 11-139.

Dst FIFO Set A-Count

Figure 11-178. DFACNT0 Register

31	30	29	28	27	26	25	24	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									ACNT																						
R-X									R-0h																						

Table 11-173. DFACNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-0	ACNT	R	0h	A-Dimension count. Number of bytes to be transferred infirst dimension.

11.8.1.2.35 DFDST0 Register (Offset = 30Ch) [reset = 0h]

DFDST0 is shown in Figure 11-179 and described in Table 11-174.

Return to the Table 11-139.

Dst FIFO Set Dst Address

Figure 11-179. DFDST0 Register

31	30	29	28	27	26	25	24	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																															

Table 11-174. DFDST0 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.

11.8.1.2.36 DFBIDX0 Register (Offset = 310h) [reset = 0h]

DFBIDX0 is shown in Figure 11-180 and described in Table 11-175.

Return to the Table 11-139.

Dst FIFO Set B-Dim Idx

Figure 11-180. DFBIDX0 Register

31	30	29	28	27	26	25	24	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															

Table 11-175. DFBIDX0 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	Src 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.

11.8.1.2.37 DFMPPRXY0 Register (Offset = 314h) [reset = X]

DFMPPRXY0 is shown in Figure 11-181 and described in Table 11-176.

Return to the Table 11-139.

Dst FIFO Set Mem Protect Proxy

Figure 11-181. DFMPPRXY0 Register

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

Table 11-176. DFMPPRXY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
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	RESERVED	R	X	
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.

11.8.1.2.38 DFBCNT0 Register (Offset = 318h) [reset = X]

DFBCNT0 is shown in Figure 11-182 and described in Table 11-177.

Return to the Table 11-139.

Dst FIFO Set B-Count

Figure 11-182. DFBCNT0 Register

31	30	29	28	27	26	25	24	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																BCNT															
R-X																R-0h															

Table 11-177. DFBCNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	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.

11.8.1.2.39 DFOPT1 Register (Offset = 340h) [reset = X]

DFOPT1 is shown in Figure 11-183 and described in Table 11-178.

Return to the Table 11-139.

Dst FIFO Set Options

Figure 11-183. DFOPT1 Register

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

Table 11-178. DFOPT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tptc_r_dbg_channel_id) and write (tptc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
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	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

Table 11-178. DFOPT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

11.8.1.2.40 DFSRC1 Register (Offset = 344h) [reset = 0h]

DFSRC1 is shown in Figure 11-184 and described in Table 11-179.

Return to the Table 11-139.

Dst FIFO Set Src Address

Figure 11-184. DFSRC1 Register

31	30	29	28	27	26	25	24	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																															

Table 11-179. DFSRC1 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.

11.8.1.2.41 DFACNT1 Register (Offset = 348h) [reset = X]

DFACNT1 is shown in Figure 11-185 and described in Table 11-180.

Return to the Table 11-139.

Dst FIFO Set A-Count

Figure 11-185. DFACNT1 Register

31	30	29	28	27	26	25	24	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									ACNT																						
R-X									R-0h																						

Table 11-180. DFACNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-0	ACNT	R	0h	A-Dimension count. Number of bytes to be transferred infirst dimension.

11.8.1.2.42 DFDST1 Register (Offset = 34Ch) [reset = 0h]

DFDST1 is shown in Figure 11-186 and described in Table 11-181.

Return to the Table 11-139.

Dst FIFO Set Dst Address

Figure 11-186. DFDST1 Register

31	30	29	28	27	26	25	24	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																															

Table 11-181. DFDST1 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.

11.8.1.2.43 DFBIDX1 Register (Offset = 350h) [reset = 0h]

DFBIDX1 is shown in Figure 11-187 and described in Table 11-182.

Return to the Table 11-139.

Dst FIFO Set B-Dim Idx

Figure 11-187. DFBIDX1 Register

31	30	29	28	27	26	25	24	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															

Table 11-182. DFBIDX1 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	Src 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.

11.8.1.2.44 DFMPPRXY1 Register (Offset = 354h) [reset = X]

DFMPPRXY1 is shown in Figure 11-188 and described in Table 11-183.

Return to the Table 11-139.

Dst FIFO Set Mem Protect Proxy

Figure 11-188. DFMPPRXY1 Register

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

Table 11-183. DFMPPRXY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
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	RESERVED	R	X	
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.

11.8.1.2.45 DFBCNT1 Register (Offset = 358h) [reset = X]

DFBCNT1 is shown in Figure 11-189 and described in Table 11-184.

Return to the Table 11-139.

Dst FIFO Set B-Count

Figure 11-189. DFBCNT1 Register

31	30	29	28	27	26	25	24	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																BCNT															
R-X																R-0h															

Table 11-184. DFBCNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	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.

12 CANFD

This chapter describes the Modular Controller Area Network (MCAN) module.

12.1 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 5 Mbit/s data rate and is compliant to ISO 11898-1:2015.

Note

The availabilty of CAN FD feature is device part number dependent. Refer to device Data Manual for more information.

shows the MCAN module overview.

12.1.1 Features

The MCAN module implements the following features:

- Conforms with ISO 11898-1:2015
- 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
- Full Message Memory capacity (4352 words).

Not supported features:

- 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)

12.2 MCAN Environment

CAN network physical layer consists of two-wire differential bus, usually twisted pair, and provides high level of interference immunity. External CAN transceiver IC is needed to access a CAN bus by the MCAN.

Figure 12-1 shows an overview of a typical MCAN application.

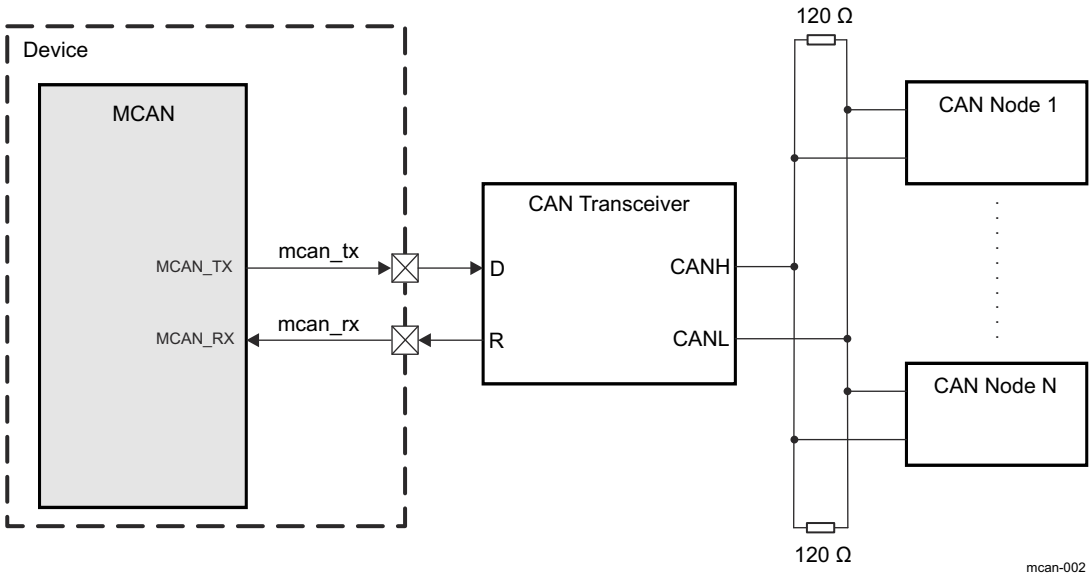


Figure 12-1. MCAN Typical Application

Table 12-1 describes the external signals of the MCAN module.

Table 12-1. MCAN I/O Description

Module Signal	Device Signal	I/O(1)	Description	Value at Reset
MCAN_RX	mcan_rx	I	Serial data input from external CAN tranceiver	HiZ
MCAN_TX	mcan_tx	O	Serial data output to external CAN tranceiver	1

(1) I = Input; O = Output

Note

The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see *Pad Configuration Registers* of *Control Module*.

12.2.1 CAN Network Basics

- CAN bus is a 2-wire differential bus using Non-Return-to-Zero (NRZ) encoding and has two states:
  - Recessive state (logical 1)
  - Dominant state (logical 0)
- The network is multicontroller. When two or more nodes (ECUs) attempt to transmit at the same time, a non-destructive arbitration technique guarantees messages are sent in order of priority and no messages are lost.
- The message transmission is multicast. Data messages transmitted are identifier based, not address based.
- Content of message is labeled by the identifier that is unique throughout the network (for example: rpm, temperature, position, pressure, and so forth).
- All nodes on network receive the message and each performs an acceptance test on the identifier. If message is relevant, it is processed, otherwise it is ignored.
- The unique identifier also determines the priority of the message (the lower the numerical value of the identifier, the higher the priority is).

- Data is transmitted and received using message frames, consisting of the following basic fields:
  - Arbitration field
  - Control field
  - Data field (up to 8 bytes for Classical CAN and up to 64 bytes for CAN FD)
  - CRC field
  - ACK field

For more information, see *ISO 11898-1:2015: CAN data link layer and physical signalling*.

12.3 MCAN Integration

MCAN Integration shows the integration of the MCAN module in the device.

Table 12-2 through Table 12-4 summarize the integration of the MCAN module in the device.

Table 12-2. MCAN Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
MCAN	PD_L4PER	Yes	L4_PER2

Table 12-3. MCAN Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MCAN	MCAN_ICLK	L4PER2_L3_GICLK/2	PRCM	Interface clock for the MCAN module
	MCAN_FCLK	MCAN_CLK	PRCM	Functional clock for the MCAN core
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MCAN	MCAN_RST	L4PER_RST	PRCM	Asynchronous reset signal to the MCAN module

Table 12-4. MCAN Hardware Requests

DMA Requests				
Module Instance	**Source Signal Name**	**DMA_CROSSBAR Input**	**Default Mapping**	**Description**
MCAN	MCAN_DREQ_TX	DMA_CROSSBAR_161	-	MCAN TX DMA Event
	MCAN_DREQ_RX_FE1	DMA_CROSSBAR_162	-	MCAN RX Filter Event 1
	MCAN_DREQ_RX_FE2	DMA_CROSSBAR_163	-	MCAN RX Filter Event 2

Note

For more information about the DMA_CROSSBAR module, see *DMA_CROSSBAR Module Functional Description* in *Control Module*.

Note

For the description of the interrupt source, see Section 12.4.2, *Interrupt and DMA Requests*.

ADVANCE INFORMATION

12.4 MCAN Functional Description

The MCAN module performs CAN protocol communication according to ISO 11898-1:2015. The bit rate can be programmed to values up to 5 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 12-2 shows the MCAN module block diagram.

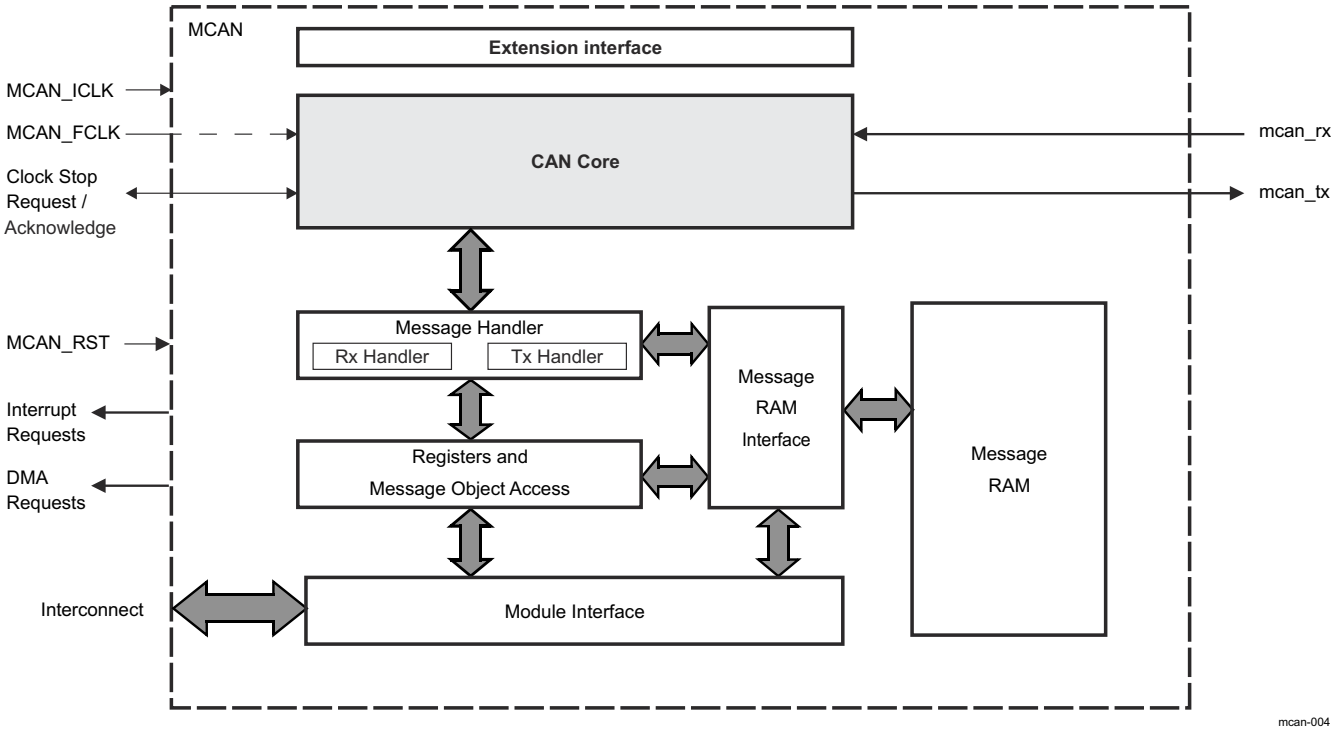


Figure 12-2. MCAN Block Diagram

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 11898-1:2015 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 12.4.11, *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.

12.4.1 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:

$f_{0,ICLK}(OCP) \pm \Delta f_{FM,ICLK}(OCP) \geq f_{FCLK}$

CAN-FD supports higher speeds of operation and as such has more stringent timing requirements than Classic CAN. For optimal performance, TI recommends using the lowest N-divider value that maintains a working PLL REF_CLK (GMAC_DSP_DPLL_CLK) for the system. Lower N-divider values increase the loop bandwidth of the PLL which in turn improves timing margins for CAN-FD.

For CAN-FD operations > 2 Mbps:

- For 20 MHz input clocks, N = 0 is the preferred configuration.

For CAN-FD operations < 2 Mbps:

- For 20 MHz input clocks, N = 0 is the preferred configuration.
- For 19.2 MHz input clocks, N = 11 is the preferred configuration.

For more information on how to configure the relevant clock source registers, see *PRCM* and the device data manual.

12.4.2 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 12.4.4.8.2, *Suspend Mode*).

12.4.2.1 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 12.4.7.2, *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 12.4.5.1, *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)

12.4.2.2 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.

For more information about available Interrupt and DMA Requests, see Section 12.3, *MCAN Integration*.

12.4.3 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.

12.4.4 Operating Modes

12.4.4.1 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.

12.4.4.2 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.

12.4.4.3 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.
- Wakeup 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.

In the CAN FD format, the DLC coding differs from the standard CAN format (see Table 12-5).

Table 12-5. DLC Coding

DLC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of Data Bytes in Standard CAN	0	1	2	3	4	5	6	7	8	8	8	8	8	8	8	8
Number of Data Bytes in CAN FD	0	1	2	3	4	5	6	7	8	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 12-3) 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.

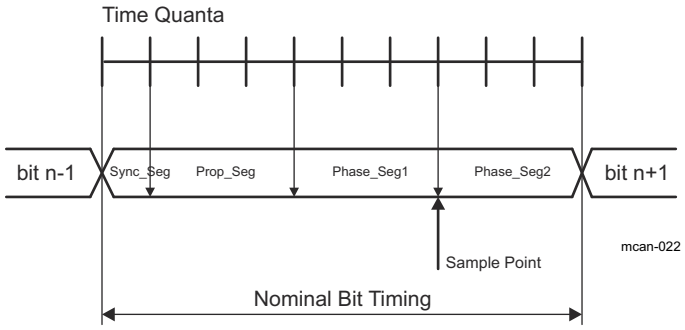


Figure 12-3. CAN Bit Timing

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.

12.4.4.4 Transmitter Delay Compensation

12.4.4.4.1 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 ISO 11898-1:2015). 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 12-4). 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.

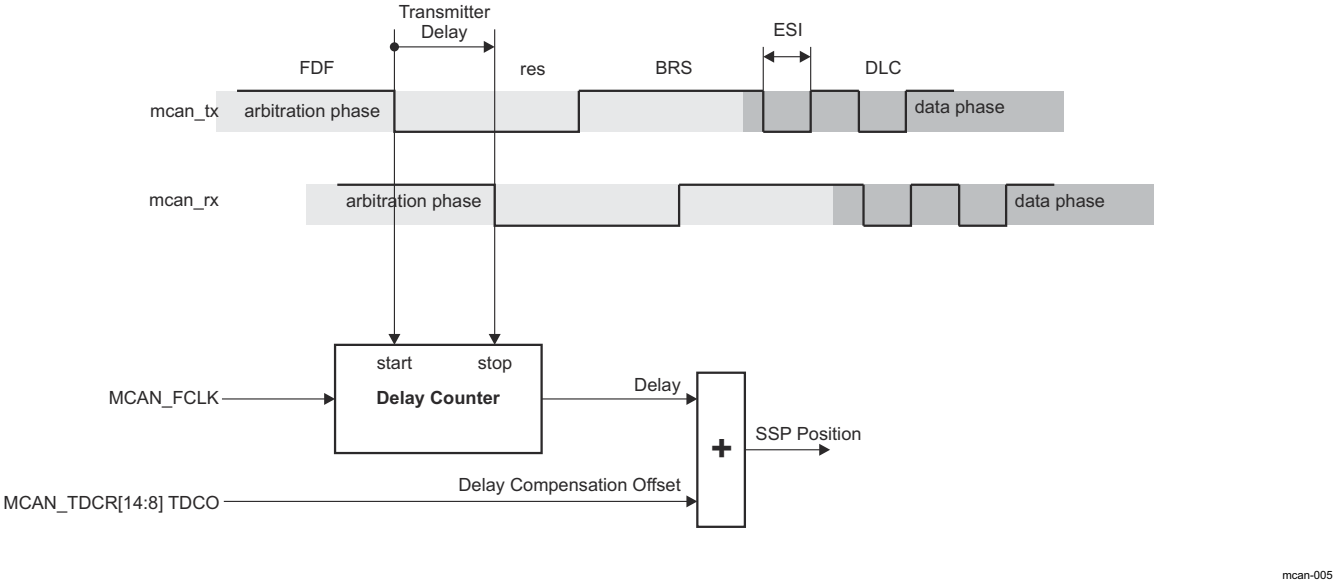


Figure 12-4. Transmitter Delay Measurement

12.4.4.4.2 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 12-4). 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.

12.4.4.5 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.

12.4.4.6 Bus Monitoring Mode

Entering Bus Monitoring Mode is done by setting the MCAN_CCCR[5] MON bit to 1. In this mode (see ISO 11898-1:2015, *Bus Monitoring* section), 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 12-5 shows the connection of the MCAN_TX and MCAN_RX signals to the MCAN module in Bus Monitoring Mode.

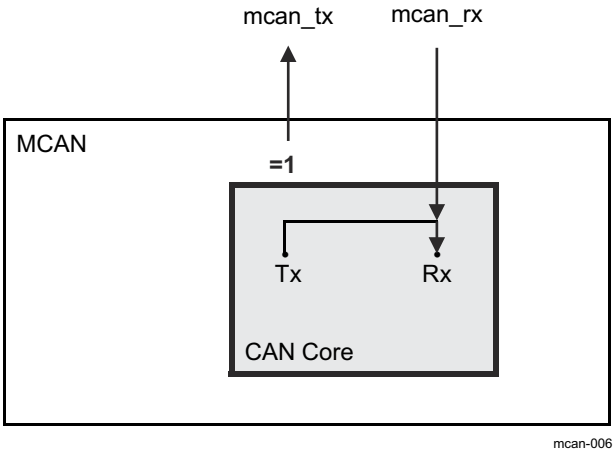


Figure 12-5. Connection of Signals in Bus Monitoring Mode

12.4.4.7 Disabled Automatic Retransmission (DAR) Mode

According to the CAN Specification (see ISO11898-1:2015, *Recovery Management* section), 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).

12.4.4.7.1 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).

12.4.4.8 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.

12.4.4.8.1 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 12.4.4.8.3, *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.

12.4.4.8.2 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 12.4.4.8.3, *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

12.4.4.8.3 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.

12.4.4.9 Test Mode

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.

12.4.4.9.1 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 12-6 shows the connection of the MCAN_TX and MCAN_RX pins to the MCAN module in case of Internal Loop Back Mode.

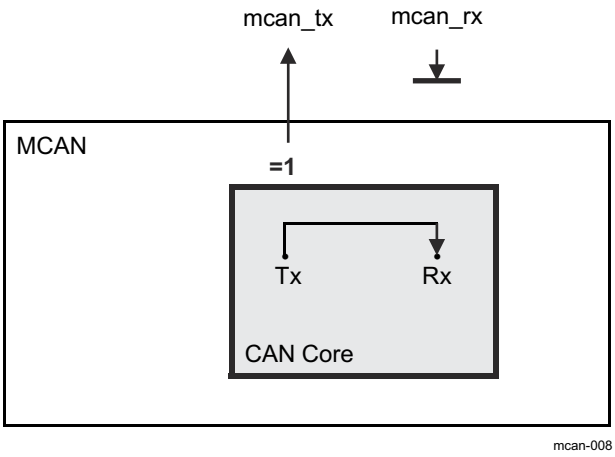


Figure 12-6. Internal Loop Back Mode

12.4.5 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 12.4.11, *Message RAM*.

12.4.5.1 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 12-7). 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.

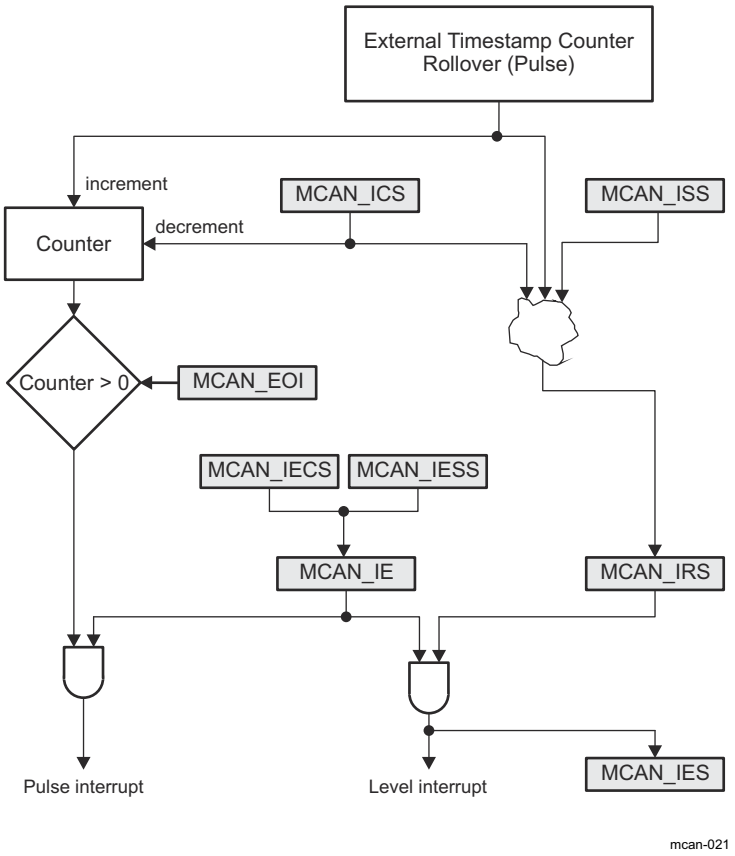


Figure 12-7. External Timestamp Counter Interrupt

12.4.6 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_CCCR[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.

12.4.7 Safety

The Message Memory is wrapped in an ECC wrapper provididing SECDED parity functionalilty. The ECC wrapper is controlled by an ECC Aggregator.

12.4.7.1 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.

12.4.7.2 ECC Aggregator

This section describes the functional details of the ECC Aggregator module.

12.4.7.2.1 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.

12.4.7.2.2 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

12.4.7.2.3 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.

12.4.7.2.4 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.

12.4.8 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

12.4.8.1 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 12.4.11, *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 12.4.2.1, *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 12.4.8.2.2 have to be considered.

12.4.8.1.1 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 12.4.11.5, *Standard Message ID Filter Element* and Section 12.4.11.6, *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.

12.4.8.1.2 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 12.4.11.5, *Standard Message ID Filter Element* and Section 12.4.11.6, *Extended Message ID Filter Element*.

12.4.8.1.3 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.

12.4.8.1.4 Standard Message ID Filtering

The standard Message ID (11-bit ID) filtering flow is shown in Figure 12-8. Section 12.4.11.5, *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

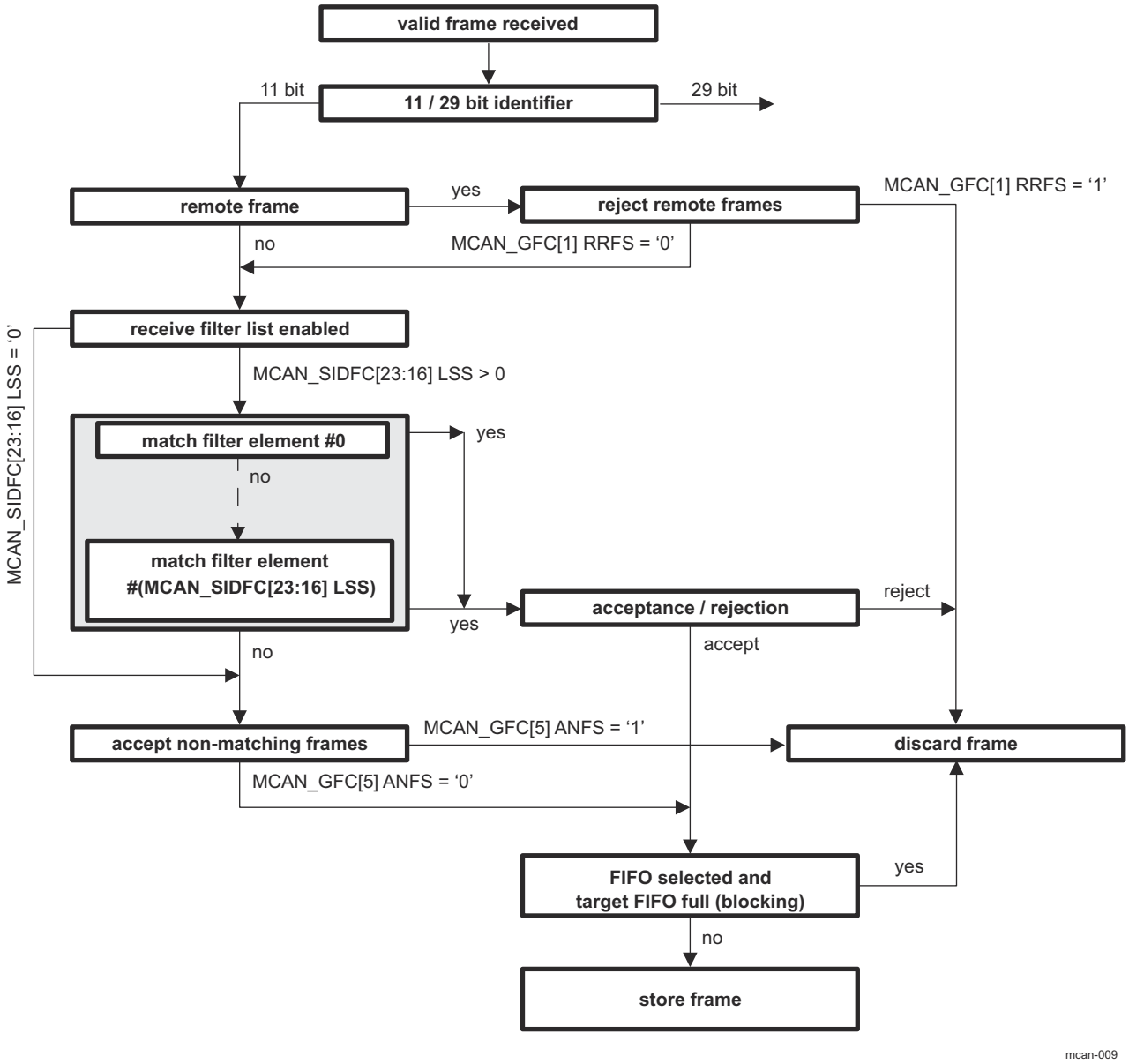


Figure 12-8. Standard Message ID Filter Path

12.4.8.1.5 Extended Message ID Filtering

The extended Message ID (29-bit ID) filtering flow is shown in Figure 12-9. Section 12.4.11.6, *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).

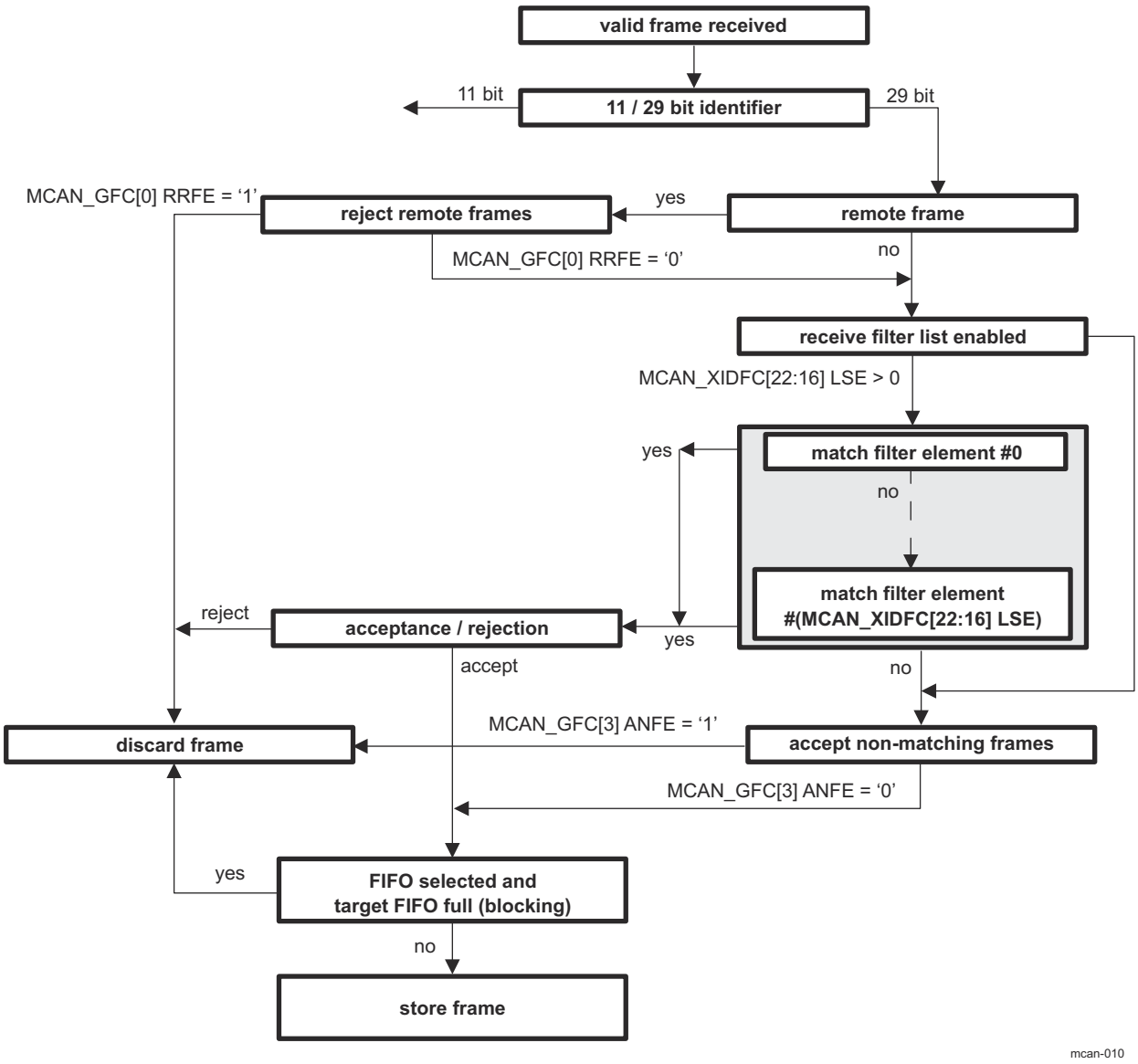


Figure 12-9. Extended Message ID Filter Path

12.4.8.2 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 12.4.8.1, *Acceptance Filtering*. Section 12.4.11.2, *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 12-10 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).

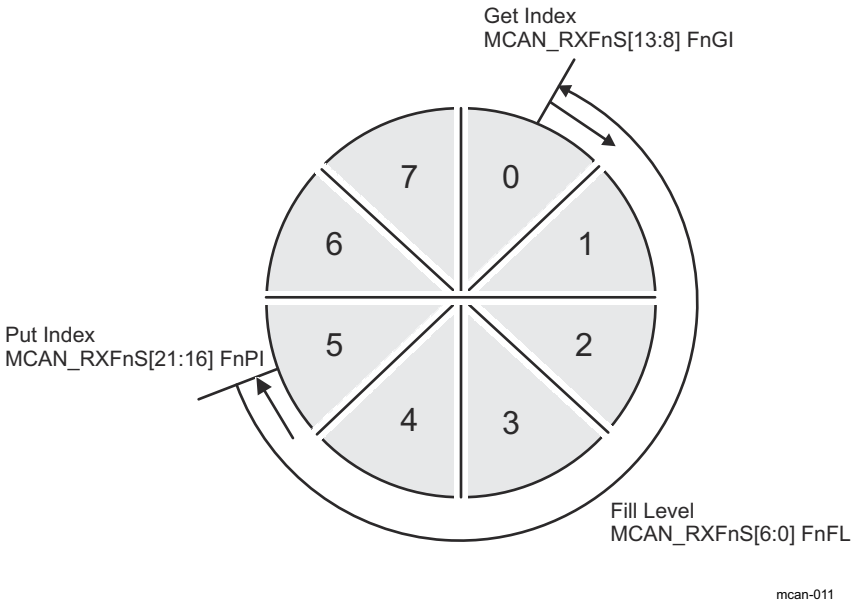


Figure 12-10. Rx FIFO Status

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 12-6 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 12-6. 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

12.4.8.2.1 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.

12.4.8.2.2 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 12-11 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.

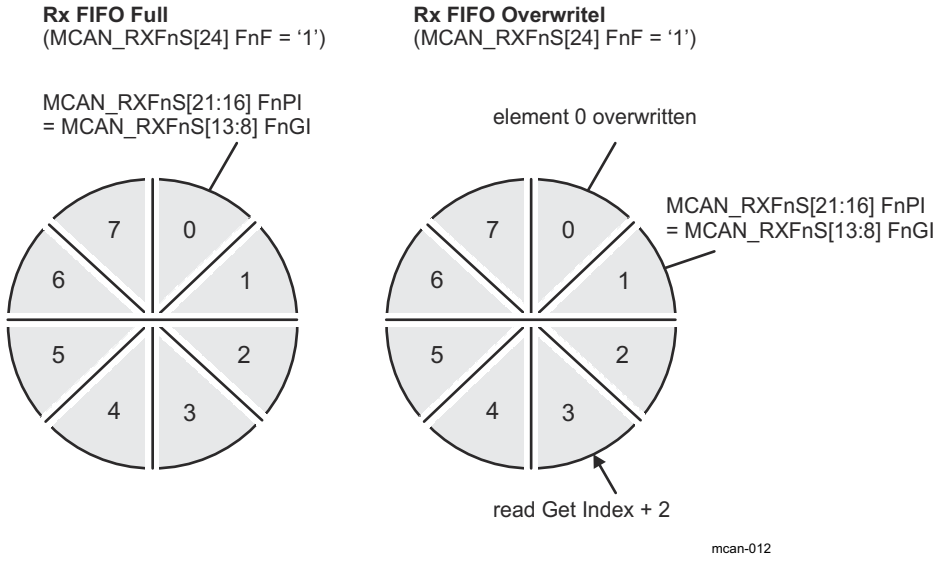


Figure 12-11. Rx FIFO Overflow Handling

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).

12.4.8.3 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 12.4.11.5, *Standard Message ID Filter Element* and Section 12.4.11.6, *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 12-7 shows Example Filter Configuration for Rx Buffers.

Table 12-7. 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

Table 12-7. Example Filter Configuration for Rx Buffers (continued)

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
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.

12.4.8.3.1 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

12.4.9 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 12.4.11.3 describes the Tx Buffer Element. Table 12-8 shows the possible configurations for message transmission.

Table 12-8. 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.

12.4.9.1 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.

12.4.9.2 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 12-9 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 12-9. 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

12.4.9.3 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 12-9). 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.

12.4.9.4 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 12-9). 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.

12.4.9.5 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 12-12 shows Mixed Dedicated Tx Buffers/Tx FIFO example.

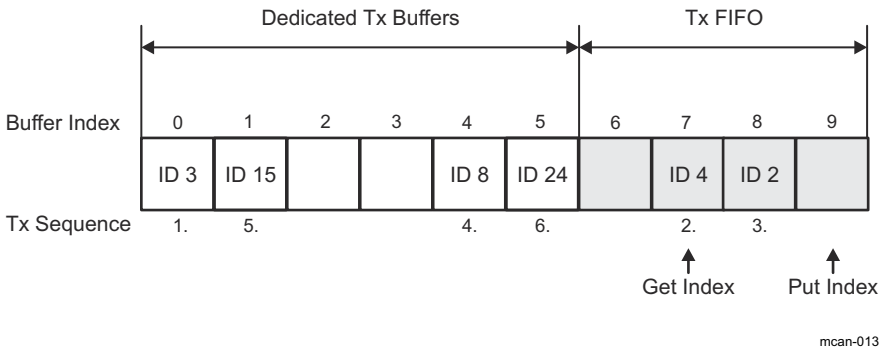


Figure 12-12. Mixed Dedicated Tx Buffers /Tx FIFO (example)

12.4.9.6 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 12-13 shows Mixed Dedicated Tx Buffers/Tx Queue example.

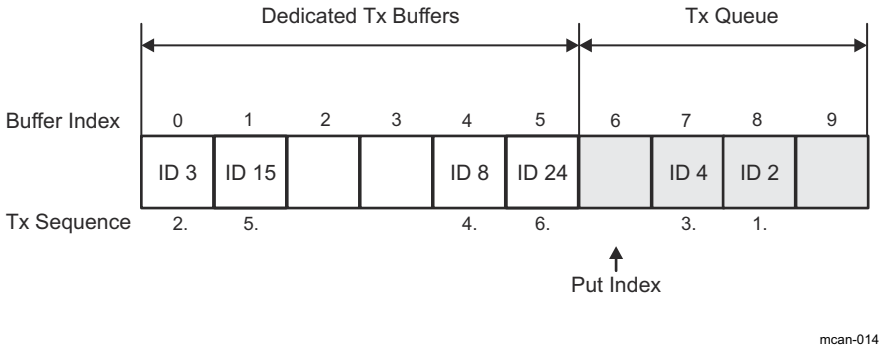


Figure 12-13. Mixed Dedicated Tx Buffers /Tx Queue (example)

12.4.9.7 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.

12.4.9.8 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 12.4.11.4 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.

12.4.10 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.

12.4.11 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

12.4.11.1 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.

Refer :#unique_380 for the address range of the Message RAM.

The Message RAM is capable to include each of the sections listed in Figure 12-14. 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

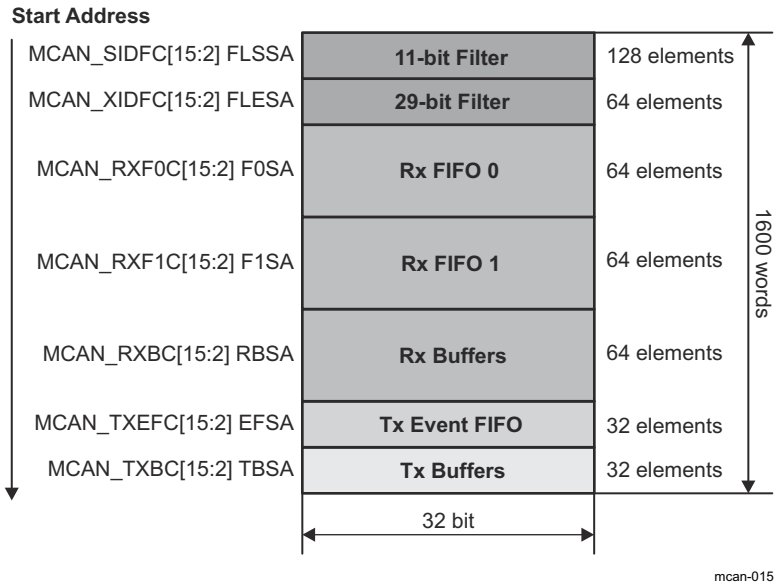


Figure 12-14. Message RAM Configuration

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

Above image is meant to allocate memory for 4352 words (max supported).

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.

12.4.11.2 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 12-15 shows Rx Buffer/Rx FIFO element structure.

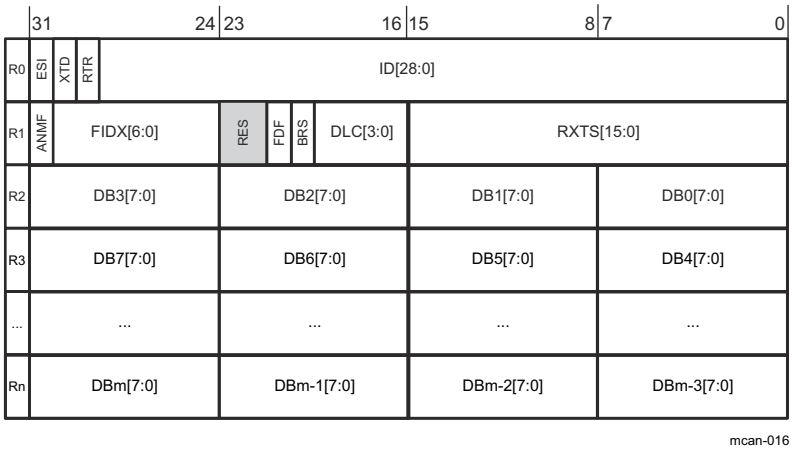


Figure 12-15. Rx Buffer/Rx FIFO Element Structure

Table 12-10 shows Rx Buffer/Rx FIFO element field descriptions.

Table 12-10. 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].

Table 12-10. Rx Buffer/Rx FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
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
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.

12.4.11.3 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 12-16 shows Tx Buffer element structure.

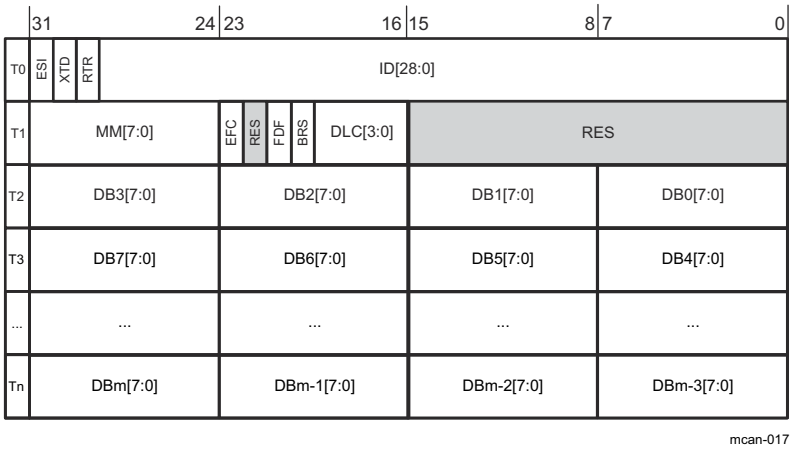


Figure 12-16. Tx Buffer Element Structure

Table 12-11 shows Tx Buffer element field descriptions.

Table 12-11. 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 ISO11898-1:2015, 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].

Table 12-11. Tx Buffer Element Field Descriptions (continued)

Word	Bits	Field Name	Description
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 12-12).
	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
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.

12.4.11.4 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 12-17 shows Tx Event FIFO element structure.

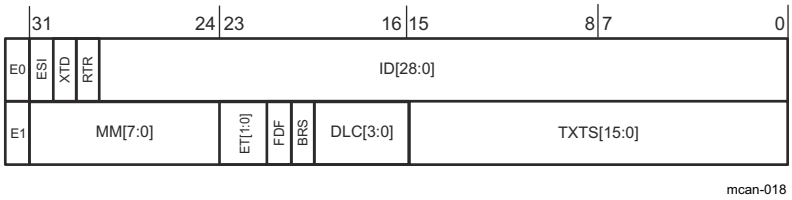


Figure 12-17. Tx Event FIFO Element Structure

Table 12-12 shows Tx Event FIFO element field descriptions.

Table 12-12. 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].

ADVANCE INFORMATION

Table 12-12. 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 12-11).
	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.

12.4.11.5 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 12-18 shows Standard Message ID Filter element structure.

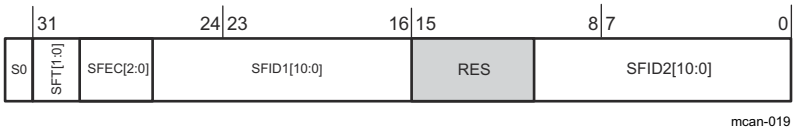


Figure 12-18. Standard Message ID Filter Element Structure

Table 12-13 shows Standard Message ID Filter element field descriptions.

ADVANCE INFORMATION

Table 12-13. 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.

12.4.11.6 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 12-19 shows Extended Message ID Filter element structure.

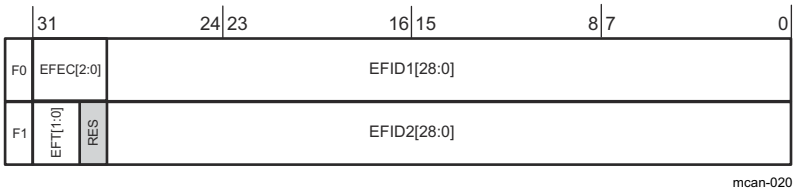


Figure 12-19. Extended Message ID Filter Element Structure

Table 12-14 shows Extended Message ID Filter element field descriptions.

Table 12-14. 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 12.4.8.1.5, *Extended Message ID Filtering*) is used.

ADVANCE INFORMATION

Table 12-14. 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.

12.5 MCAN Smart Idle Implementation

The following diagram describes the Smart Idle Implementation for MCAN.

Device supports two modes of Smart IDLE.

- Manual Mode - In this mode of operation, SMART_IDLE_WAKE_AUTO_EN = 0. The control bit is directly connected to the CAN_CLKSTOP_REQ. The entry and exit to Smart Idle is user controlled based on polling SMART_IDLE_ACK and SMART_IDLE_WAKEUP.
- Automatic Mode - In this mode of operation, entry to smart idle mode is manual by setting SMART_IDLE_ENABLE = 1. When the clkstop_wakeup signal from MCAN is asserted (based on the activity), the clkstop_req is pulled low automatically.

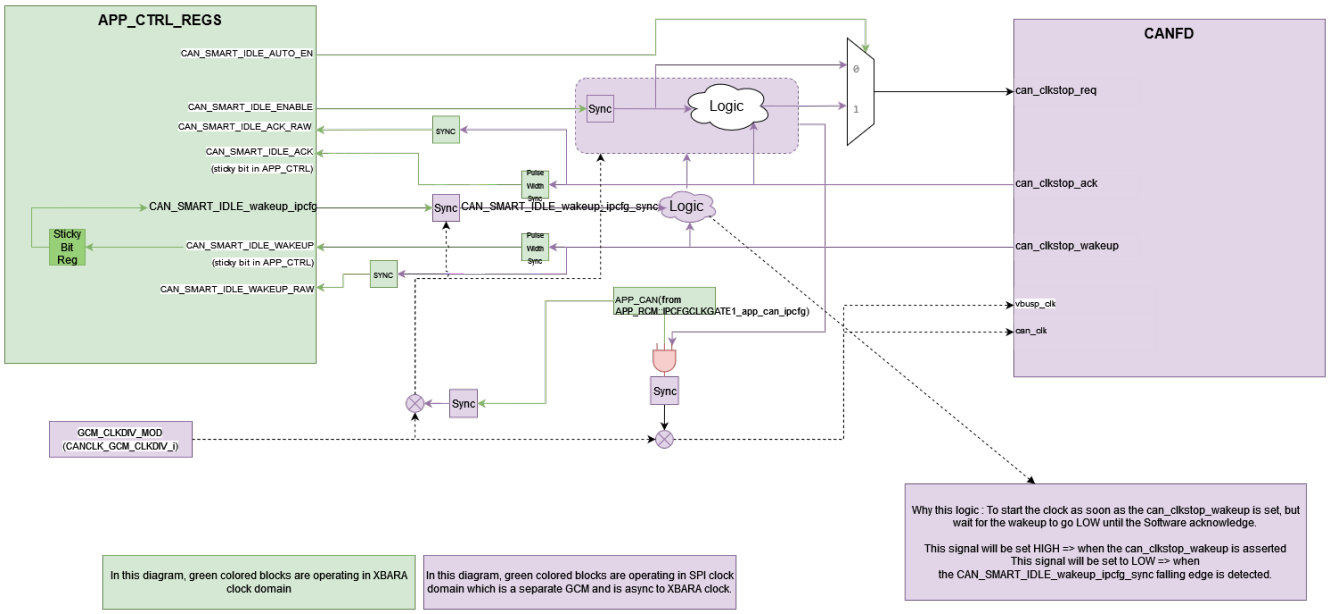


Figure 12-20. MCAN Smart Idle

The table below shows the truth table of the logic shown in cloud in the above diagram.

SMART_IDLE_ENABLE	CLKSTOP_WAKEUP	CLKSTOP_ACK	CLKSTOP_REQ (AUTO_WAKEUP=1)	ICG_EN (AUTO_WAKEUP=1/0)
0	0	0	Old State	1
0	0	1	Old State	1
0	1 (valid is slave mode)	0	Old State	1
0	1 (valid is slave mode)	1	Old State	1
1	0	0	1	1
1	**0**	**1**	**1**	**0 = clk shut**
1	1 (valid is slave mode)	0	0	1
1	1 (valid is slave mode)	1	0	1

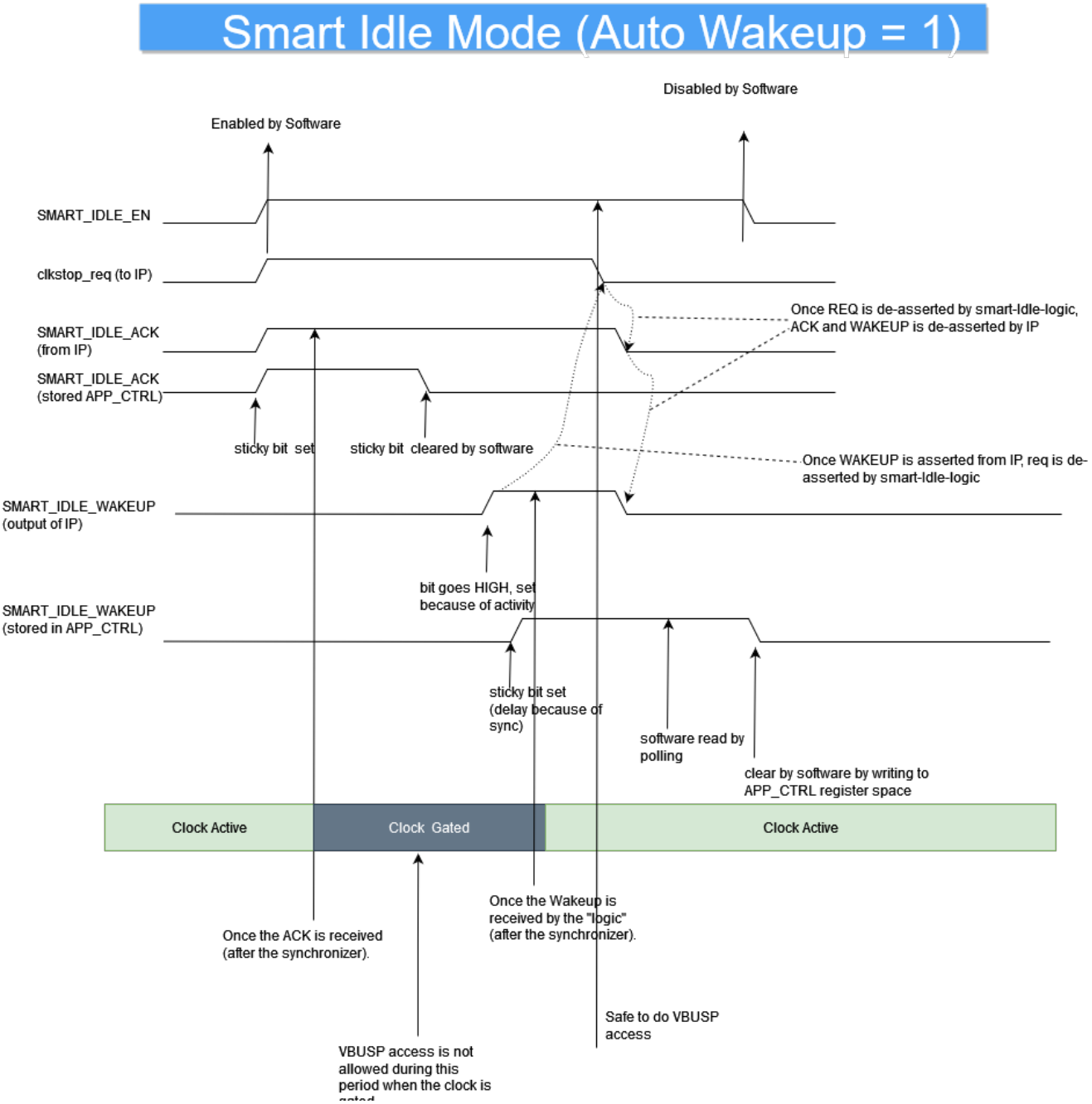


Figure 12-21. CAN Smart Idle Timing Diagram

12.5.1 Programming Sequence

Auto Wakeup = 1

- Configure CANFD as required
- Enable SmartIdle (by setting CAN_SMART_IDLE_ENABLE=1) after ensuring that there is **no** pending transaction from/to CANFD or any more access to be done to CANFD by CPU or DMA
- If any register or memory access to CANFD has to be done, disable SmartIDLE mode (by setting CAN_SMART_IDLE_ENABLE=0)
- If there is wakeup from CANFD (because of dominant bit transitioning to LOW), smart-Idle will be .
- Disable SmartIdle configuration (by setting CAN_SMART_IDLE_ENABLE=0) - to do the register access.

12.6 MCAN (APP_CANCFG) Register Manual (Base Address- 0x53F7F800)

Note

After hardware reset, the registers of the MCAN module hold the values shown in the register descriptions.

Additionally, the Bus_Off state is reset and the MCAN_TX pin is set to recessive (high). The MCAN_CCCR[0] INIT bit is set to enable the software initialization. The MCAN module will not influence the CAN bus until the software resets the MCAN_CCCR[0] INIT bit.

12.6.1 MCAN Register Summary

Table 12-15. MCAN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset
MCANSS_PID	R	32	0x0000 1900
MCANSS_CTRL	RW	32	0x0000 1904
MCANSS_STAT	R	32	0x0000 1908
MCANSS_ICS	RW	32	0x0000 190C
MCANSS_IRS	RW	32	0x0000 1910
MCANSS_IECS	RW	32	0x0000 1914
MCANSS_IE	RW	32	0x0000 1918
MCANSS_IES	R	32	0x0000 191C
MCANSS_EOI	RW	32	0x0000 1920
MCANSS_EXT_TS_PRESCALER	RW	32	0x0000 1924
MCANSS_EXT_TS_UNSERVICED_INTR_CNTR	R	32	0x0000 1928
MCANSS_ECC_EOI	RW	32	0x0000 1980
MCAN_CREL	R	32	0x0000 1A00
MCAN_ENDN	R	32	0x0000 1A04
RESERVED	R	32	0x0000 1A08
MCAN_DBTP	RW	32	0x0000 1A0C
MCAN_TEST	RW	32	0x0000 1A10
MCAN_RWD	RW	32	0x0000 1A14
MCAN_CCCR	RW	32	0x0000 1A18
MCAN_NBTP	RW	32	0x0000 1A1C
MCAN_TSCC	RW	32	0x0000 1A20
MCAN_TSCV	RW	32	0x0000 1A24
MCAN_TOCC	RW	32	0x0000 1A28
MCAN_TOCV	RW	32	0x0000 1A2C
RESERVED	R	32	0x0000 1A30
RESERVED	R	32	0x0000 1A34
RESERVED	R	32	0x0000 1A38
RESERVED	R	32	0x0000 1A3C
MCAN_ECR	R	32	0x0000 1A40
MCAN_PSR	R	32	0x0000 1A44
MCAN_TDCR	RW	32	0x0000 1A48
RESERVED	R	32	0x0000 1A4C
MCAN_IR	RW	32	0x0000 1A50
MCAN_IE	RW	32	0x0000 1A54
MCAN_ILS	RW	32	0x0000 1A58
MCAN_ILE	RW	32	0x0000 1A5C
RESERVED	R	32	0x0000 1A60

Table 12-15. MCAN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset
RESERVED	R	32	0x0000 1A64
RESERVED	R	32	0x0000 1A68
RESERVED	R	32	0x0000 1A6C
RESERVED	R	32	0x0000 1A70
RESERVED	R	32	0x0000 1A74
RESERVED	R	32	0x0000 1A78
RESERVED	R	32	0x0000 1A7C
MCAN_GFC	RW	32	0x0000 1A80
MCAN_SIDFC	RW	32	0x0000 1A84
MCAN_XIDFC	RW	32	0x0000 1A88
RESERVED	R	32	0x0000 1A8C
MCAN_XIDAM	RW	32	0x0000 1A90
MCAN_HPMS	R	32	0x0000 1A94
MCAN_NDAT1	RW	32	0x0000 1A98
MCAN_NDAT2	RW	32	0x0000 1A9C
MCAN_RXF0C	RW	32	0x0000 1AA0
MCAN_RXF0S	R	32	0x0000 1AA4
MCAN_RXF0A	RW	32	0x0000 1AA8
MCAN_RXBC	RW	32	0x0000 1AAC
MCAN_RXF1C	RW	32	0x0000 1AB0
MCAN_RXF1S	R	32	0x0000 1AB4
MCAN_RXF1A	RW	32	0x0000 1AB8
MCAN_RXESC	RW	32	0x0000 1ABC
MCAN_TXBC	RW	32	0x0000 1AC0
MCAN_TXFQS	R	32	0x0000 1AC4
MCAN_TXESC	RW	32	0x0000 1AC8
MCAN_TXBRP	R	32	0x0000 1ACC
MCAN_TXBAR	RW	32	0x0000 1AD0
MCAN_TXBCR	RW	32	0x0000 1AD4
MCAN_TXBTO	R	32	0x0000 1AD8
MCAN_TXBCF	R	32	0x0000 1ADC
MCAN_TXBTIE	RW	32	0x0000 1AE0
MCAN_TXBCIE	RW	32	0x0000 1AE4
RESERVED	R	32	0x0000 1AE8
RESERVED	R	32	0x0000 1AEC
MCAN_TXEFC	RW	32	0x0000 1AF0
MCAN_TXEFS	R	32	0x0000 1AF4
MCAN_TXEFA	RW	32	0x0000 1AF8
RESERVED	R	32	0x0000 1AFC
MCANSS_ECC_AGGR_REVISION	R	32	0x0000 1C00
MCANSS_ECC_VECTOR	RW	32	0x0000 1C08
MCANSS_ECC_MISC_STATUS	R	32	0x0000 1C0C
MCANSS_ECC_WRAP_REVISION	R	32	0x0000 1C10
MCANSS_ECC_CONTROL	RW	32	0x0000 1C14
MCANSS_ECC_ERR_CTRL1	RW	32	0x0000 1C18
MCANSS_ECC_ERR_CTRL2	RW	32	0x0000 1C1C
MCANSS_ECC_ERR_STAT1	RW	32	0x0000 1C20

Table 12-15. MCAN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset
MCANSS_ECC_ERR_STAT2	R	32	0x0000 1C24
MCANSS_ECC_SEC_EOI_REG	RW	32	0x0000 1C3C
MCANSS_ECC_SEC_STATUS_REG0	RW	32	0x0000 1C40
MCANSS_ECC_SEC_ENABLE_SET_REG0	RW	32	0x0000 1C80
MCANSS_ECC_SEC_ENABLE_CLR_REG0	RW	32	0x0000 1CC0
MCANSS_ECC_DED_EOI_REG	RW	32	0x0000 1D3C
MCANSS_ECC_DED_STATUS_REG0	RW	32	0x0000 1D40
MCANSS_ECC_DED_ENABLE_SET_REG0	RW	32	0x0000 1D80
MCANSS_ECC_DED_ENABLE_CLR_REG0	RW	32	0x0000 1DC0

12.6.2 MCAN Register Description

Table 12-16. MCANSS_PID

Address Offset	0x0000 1900
Description	Revision Register The Revision Register contains the major and minor revisions for the module.
Type	R

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

Bits	Field Name	Description	Type	Reset
31:0	REVISION	MCAN revision version	R	0x-

Table 12-17. MCANSS_CTRL

Address Offset	0x0000 1904
Description	Control Register The Control Register contains general control bits for the MCAN module.
Type	RW

31	30	29	28	27	26	25	24	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																									EXT_TS_CNTR_EN	AUTOWAKEUP	WAKEUPREQEN	FREE	SOFT	CLKFACK	RESET

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6	EXT_TS_CNTR_EN	External Timestamp Counter Enable	RW	0x0
5	AUTOWAKEUP	Automatic Wakeup Enable	RW	0x0
4	WAKEUPREQEN	Wakeup Request Enable	RW	0x0
3	FREE	0x0: Disregard debug suspend 0x1: Enable Debug Suspend	RW	0x1
2	SOFT	If FREE = 0x1: 0x0: debug suspend doesn't wait for Idle 0x1: debug suspend waits for Idle	RW	0x0
1	CLKFACK	Clock Fast Ack	RW	0x0

Bits	Field Name	Description	Type	Reset
0	RESET	Initiates a Soft Reset **Note**: Software application should complete all pending MCAN services before applying the soft reset. Accesses to MCAN core registers will be stalled until soft reset is completed.	W	0x0

Table 12-18. MCANSS_STAT

Address Offset 0x0000 1908

Description Status Regsiter
The Status register provide general status bits for the MCAN module.

Type R

31	30	29	28	27	26	25	24	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																								STATE					ENABLE_FDOE	MEM_INIT_DONE	RESET

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:3	STATE	0x0: Active 0x1: In transition to Idle 0x2: Idle 0x3: In transition to Active	R	0x0
2	ENABLE_FDOE	Enable CAN FD configuration	R	0x-
1	MEM_INIT_DONE	0x0: Memory Initialization is in progress 0x1: Memory Intialization Done	R	0x0
0	RESET	0x0: Not in reset 0x1: Reset is in progress	R	0x0

Table 12-19. MCANSS_ICS

Address Offset 0x0000 190C

Description Interrupt Clear Shadow Register
Write to clear interrupt bits.

Type RW

31	30	29	28	27	26	25	24	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																															EXT_TS_CNTR_OVFL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt status. Write 1 to clear bits.	W	0x0

Table 12-20. MCANSS_IRS

Address Offset 0x0000 1910

Table 12-20. MCANSS_IRS (continued)

Description Interrupt Raw Status Register
Read raw interrupt status. Write 1 to set interrupt bits.

Type RW

31	30	29	28	27	26	25	24	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																															EXT_TS_CNTR_OVFL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt status.	RW1TS	0x0

Table 12-21. MCANSS_IECS

Address Offset 0x0000 1914

Description Interrupt Enable Clear Shadow Register
Write to clear interrupt enable bits.

Type RW

31	30	29	28	27	26	25	24	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																															EXT_TS_CNTR_OVFL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt. Write 1 to clear bits.	W	0x0

Table 12-22. MCANSS_IE

Address Offset 0x0000 1918

Description Interrupt Enable Register
Read interrupt Enable.

Type RW

31	30	29	28	27	26	25	24	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																															EXT_TS_CNTR_OVFL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt.	RW1TS	0x0

Table 12-23. MCANSS_IES

Address Offset	0x0000 191C
Description	Interrupt Enable Status Read Enabled Interrupts.
Type	R

31	30	29	28	27	26	25	24	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																															EXT_TS_CNTR_OVFL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt.	R	0x0

Table 12-24. MCANSS_EOI

Address Offset	0x0000 1920
Description	End Of Interrupt End of Interrupt Register.
Type	RW

31	30	29	28	27	26	25	24	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																								EOI							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:0	EOI	Write with bit position of targetted interrupt (example: External TS is bit 0). Upon write, level interrupt will clear and if unserviced interrupt counter > 1 will issue another pulse interrupt.	W	0x0

Table 12-25. MCANSS_EXT_TS_PRESCALER

Address Offset	0x0000 1924
Description	External Timestamp PreScaler 0 External TImeStamp PreScaler.
Type	RW

31	30	29	28	27	26	25	24	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								PRESCALER																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:0	PRESCALER	External Timestamp Prescaler reload value. External Timestamp count rate is Host clock (MCAN_ICLK) rate divided by this vlaue.	RW	0x0

Table 12-26. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR

Address Offset	0x0000 1980

ADVANCE INFORMATION

Table 12-26. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR (continued)

Description External Timestamp PreScaler 0 Unserviced Interrupts Counter
External TImeStamp Unserviced Interrupts Counter.

Type R

31	30	29	28	27	26	25	24	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																														EXT_TS_INTR_CNTR	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4:0	EXT_TS_INTR_CNTR	Number of unserviced rollover interupts. If > 1 an EOI write will issue another pulse interrupt.	R	0x0

Table 12-27. MCANSS_ECC_EOI

Address Offset 0x0000 1980

Description ECC EOI
End Of Interrupt for ECC interrupt.

Type RW

31	30	29	28	27	26	25	24	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																							ECC_EOI	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8	ECC_EOI	ECC EOI	W	0x0
7:0	RESERVED	Reserved	R	0x0

Table 12-28. MCAN_CREL

Address Offset 0x0000 1A00

Description Core Release Register
Release dependent constant (version + date).

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REL				STEP				SUBSTEP				YEAR				MON								DAY							

Bits	Field Name	Description	Type	Reset
31:28	REL	Core Release One digit, BCD-coded.	R	0x3
27:24	STEP	Step of Core Release One digit, BCD-coded.	R	0x2
23:20	SUBSTEP	Sub-step of Core Release One digit, BCD-coded.	R	0x1
19:16	YEAR	Time Stamp Year One digit, BCD-coded.	R	0x5
15:8	MON	Time Stamp Month Two digits, BCD-coded.	R	0x3
7:0	DAY	Time Stamp Day Two digits, BCD-coded.	R	0x20

Table 12-29. MCAN_ENDN

Address Offset	0x0000 1A04
Description	Endian Register Constant 0x8765 4321.
Type	R

31	30	29	28	27	26	25	24	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																															

Bits	Field Name	Description	Type	Reset
31:0	ETV	Endianness Test Value The endianness test value is 0x8765 4321.	R	0x8765 4321

Table 12-30. MCAN_DBTP

Address Offset	0x0000 1A0C
Description	Data Bit Timing & Prescaler Register Configuration of data phase bit timing, transmitter delay compensation enable. This register is only writable if the MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set. The CAN bit time may be programed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 MCAN_FCLK periods. t_q = (MCAN_DBTP[20:16] DBRP + 1) mtq (minimum time quantum = CAN clock period (MCAN_FCLK)). The MCAN_DBTP[12:8] DTSEG1 field is the sum of Prop_Seg and Phase_Seg1. The MCAN_DBTP[7:4] DTSEG2 field is Phase_Seg2. Therefore the length of the bit time is (programmed values) [MCAN_DBTP[12:8] DTSEG1 + MCAN_DBTP[7:4] DTSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q. The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point. **Note:** With a CAN clock (MCAN_FCLK) of 8 MHz, the reset value of 0x0000 0A33 configures the MCAN module for a data phase bit rate of 500 kBit/s. **Note:** The bit rate configured for the CAN FD data phase via the MCAN_DBTP register must be higher or equal to the bit rate configured for the arbitration phase via the MCAN_NBTP register.
Type	RW

31	30	29	28	27	26	25	24	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								TDC	RESERVED		DBRP					RESERVED			DTSEG1					DTSEG2				DSJW			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23	TDC	Transmitter Delay Compensation 0x0: Transmitter Delay Compensation disabled 0x1: Transmitter Delay Compensation enabled	RW	0x0
22:21	RESERVED	Reserved	R	0x0
20:16	DBRP	Data Baud Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 31 (0x00-0x1F). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x0
15:13	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
12:8	DTSEG1	Data time segment before sample point Valid values are 0 to 31 (0x00-0x1F). The actual interpretation by the hardware of this value is such that one more than the programmed value is used.	RW	0xA
7:4	DTSEG2	Data time segment after sample point Valid values are 0 to 15 (0x0-0xF). The actual interpretation by the hardware of this value is such that one more than the programmed value is used.	RW	0x3
3:0	DSJW	Data (Re)Synchronization Jump Width Valid values are 0 to 15 (0x0-0xF). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x3

Table 12-31. MCAN_TEST

Address Offset 0x0000 1A10

Description Test Register
Test mode selection.
Write access to the Test Register has to be enabled by setting the MCAN_CCCR[7] TEST bit. All Test Register functions are set to their reset values when the MCAN_CCCR[7] TEST bit is reset.
Loop Back Mode and software control of the MCAN_TX pin are hardware test modes. Programming of the MCAN_TEST[6:5] TX field ≠ 00 may disturb the message transfer on the CAN bus.

Type RW

31	30	29	28	27	26	25	24	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																								RX	TX		LBCK	RESERVED			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	RX	Receive Pin Monitors the actual value of the MCAN_RX pin 0x0: The CAN bus is dominant (MCAN_RX = 0) 0x1: The CAN bus is recessive (MCAN_RX = 1)	R	0x0
6:5	TX	Control of Transmit Pin 0x0: Reset value, the MCAN_TX pin controlled by the CAN Core, updated at the end of the CAN bit time 0x1: Sample Point can be monitored at the MCAN_TX pin 0x2: Dominant (0) level at the MCAN_TX pin 0x3: Recessive (1) at the MCAN_TX pin	RW	0x0
4	LBCK	Loop Back Mode 0x0: Reset value, Loop Back Mode is disabled 0x1: Loop Back Mode is enabled (see Section 12.4.4.9, *Test Modes*)	RW	0x0
3:0	RESERVED	Reserved	R	0x0

Table 12-32. MCAN_RWD

Address Offset 0x0000 1A14

Table 12-32. MCAN_RWD (continued)

Description	RAM Watchdog Monitors the READY output of the Message RAM. The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access starts the Message RAM Watchdog Counter with the value configured by the MCAN_RWD[7:0] WDC field. The counter is reloaded with the MCAN_RWD[7:0] WDC field when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR[26] WDI is set. The RAM Watchdog Counter is clocked by the Host clock (MCAN_ICLK).
Type	RW

31	30	29	28	27	26	25	24	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																WDV								WDC							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:8	WDV	Watchdog Value Actual Message RAM Watchdog Counter Value.	R	0x0
7:0	WDC	Watchdog Configuration Start value of the Message RAM Watchdog Counter. With the reset value of 00 the counter is disabled.	RW	0x0

Table 12-33. MCAN_CCCR

Address Offset	0x0000 1A18
Description	CC Control Register Operation mode configuration. For details about setting and resetting of single bits, see Section 12.4.4.1, *Software Initialization*.
Type	RW

31	30	29	28	27	26	25	24	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																TXP	EFBI	PXHD	RESERVED			BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x0
14	TXP	Transmit Pause If this bit is set, the MCAN module pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see Section 12.4.9, *Tx Handling*) 0x0: Transmit pause disabled 0x1: Transmit pause enabled	RW	0x0
13	EFBI	Edge Filtering during Bus Integration 0x0: Edge filtering disabled 0x1: Two consecutive dominant t_q required to detect an edge for hard synchronization	RW	0x0
12	PXHD	Protocol Exception Handling Disable 0x0: Protocol exception handling enabled 0x1: Protocol exception handling disabled **Note:** When protocol exception handling is disabled, the MCAN module will transmit an error frame when it detects a protocol exception condition.	RW	0x0
11:10	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
9	BRSE	Bit Rate Switch Enable 0x0: Bit rate switching for transmissions disabled 0x1: Bit rate switching for transmissions enabled **Note:** When CAN FD operation is disabled the MCAN_CCCR[8] FDOE = 0, the MCAN_CCCR[9] BRSE bit is not evaluated.	RW	0x0
8	FDOE	FD Operation Enable 0x0: FD operation disabled 0x1: FD operation enabled	RW	0x0
7	TEST	Test Mode Enable 0x0: Normal operation. The MCAN_TEST register holds reset values 0x1: Test Mode. Write access to the MCAN_TEST register enabled	RW	0x0
6	DAR	Disable Automatic Retransmission 0x0: Automatic retransmission of messages not transmitted successfully enabled 0x1: Automatic retransmission disabled	RW	0x0
5	MON	Bus Monitoring Mode The MCAN_CCCR[5] MON bit can only be set by the Host CPU when both MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set to 1. The bit can be reset by the Host CPU at any time. 0x0: Bus Monitoring Mode is disabled 0x1: Bus Monitoring Mode is enabled	RW	0x0
4	CSR	Clock Stop Request 0x0: No clock stop is requested 0x1: Clock stop requested. When clock stop is requested, first the MCAN_CCCR[0] INIT bit and then the MCAN_CCCR[3] CSA bit will be set after all pending transfer requests have been completed and the CAN bus reached idle.	RW	0x0
3	CSA	Clock Stop Acknowledge 0x0: No clock stop acknowledged 0x1: The MCAN module may be set in power down by stopping MCAN_ICLK and MCAN_FCLK	R	0x0
2	ASM	Restricted Operation Mode The MCAN_CCCR[2] ASM bit can only be set by the Host CPU when both MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set to 1. The bit can be reset by the Host CPU at any time. For a description of the Restricted Operation Mode, see Section 12.4.4.5. 0x0: Normal CAN operation 0x1: Restricted Operation Mode active	RW	0x0
1	CCE	Configuration Change Enable 0x0: The Host CPU has no write access to the protected configuration registers 0x1: The Host CPU has write access to the protected configuration registers (while the MCAN_CCCR[0] INIT = 1)	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INIT	Initialization 0x0: Normal Operation 0x1: Initialization is started **Note:** Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to the MCAN_CCCR[0] INIT bit can be read back. Therefore the software has to assure that the previous value written to the MCAN_CCCR[0] INIT bit has been accepted by reading the MCAN_CCCR[0] INIT bit before setting the MCAN_CCCR[0] INIT bit to a new value.	RW	0x1

Table 12-34. MCAN_NBTP

Address Offset 0x0000 1A1C

Description Nominal Bit Timing & Prescaler Register
Configuration of arbitration phase bit timing.
This register is only writable if the MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set. The CAN bit time may be programed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 MCAN_FCLK periods. t_q = (MCAN_NBTP[24:16] NBRP + 1) mtq. The MCAN_NBTP[15:8] NTSEG1 field is the sum of Prop_Seg and Phase_Seg1. The MCAN_NBTP[6:0] NTSEG2 field is Phase_Seg2.
Therefore the length of the bit time is (programmed values) [MCAN_NBTP[15:8] NTSEG1 + MCAN_NBTP[6:0] NTSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSJW								NBRP								NTSEG1								RESERVED	NTSEG2						

Bits	Field Name	Description	Type	Reset
31:25	NSJW	Nominal (Re)Synchronization Jump Width Valid values are 0 to 127 (0x00-0x7F). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x3
24:16	NBRP	Nominal Baud Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 511 (0x000-0x1FF). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x0
15:8	NTSEG1	Nominal Time segment before sample point Valid values are 1 to 255 (0x01-0xFF). The actual interpretation by the hardware of this value is such that one more than the programmed value is used.	RW	0xA
7	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
6:0	NTSEG2	Nominal Time segment after sample point Valid values are 0 to 127 (0x00-0x7F). The actual interpretation by the hardware of this value is such that one more than the programmed value is used. **Note:** With a CAN clock (MCAN_FCLK) of 8 MHz, the reset value of 0x0600 0A03 configures the MCAN module for a bit rate of 500 kBit/s.	RW	0x3

Table 12-35. MCAN_TSCC

Address Offset	0x0000 1A20
Description	Timestamp Counter Configuration Timestamp counter prescaler setting, selection of internal/external timestamp vector. For a description of the Timestamp Counter, see Section 12.4.5, *Timestamp Generation*.
Type	RW

31	30	29	28	27	26	25	24	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												TCP				RESERVED														TSS	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x0
19:16	TCP	Timestamp Counter Prescaler Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1-16 (0x0-0xF)]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. **Note:** With CAN FD an external counter is required for timestamp generation (MCAN_TSCC[1:0] TSS = 10)	RW	0x0
15:2	RESERVED	Reserved	R	0x0
1:0	TSS	Timestamp Select 0x0: Timestamp counter value always 0x0000 0x1: Timestamp counter value incremented according to the MCAN_TSCC[19:16] TCP field 0x2: External timestamp counter value used 0x3: Same as 00	RW	0x0

Table 12-36. MCAN_TSCV

Address Offset	0x0000 1A24
Description	Timestamp Counter Value Read/reset timestamp counter.
Type	RW

31	30	29	28	27	26	25	24	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																TSC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
15:0	TSC	Timestamp Counter The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx). When the MCAN_TSCC[1:0] TSS = 01, the Timestamp Counter is incremented in multiples of CAN bit times [1-16] depending on the configuration of the MCAN_TSCC[19:16] TCP field. A wrap around sets interrupt flag MCAN_IR[16] TSW. Write access resets the counter to zero. When the MCAN_TSCC[1:0] TSS = 10, the MCAN_TSCV[15:0] TSC field reflects the external Timestamp Counter value. A write access has no impact. **Note:** A 'wrap around' is a change of the Timestamp Counter value from non-zero to zero not caused by write access to the MCAN_TSCV register.	RWTC	0x0

Table 12-37. MCAN_TOCC

Address Offset	0x0000 1A28
Description	Timeout Counter Configuration Configuration of timeout period, selection of timeout counter operation mode. For a description of the Timeout Counter, see Section 12.4.6, *Timeout Counter*.
Type	RW

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

Bits	Field Name	Description	Type	Reset
31:16	TOP	Timeout Period Start value of the Timeout Counter (down-counter). Configures the Timeout Period.	RW	0xFFFF
15:3	RESERVED	Reserved	R	0x0
2:1	TOS	Timeout Select When operating in Continuous mode, a write to the MCAN_TOCV[15:0] TOC field presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field and continues down-counting. When 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. 0x0: Continuous operation 0x1: Timeout controlled by Tx Event FIFO 0x2: Timeout controlled by Rx FIFO 0 0x3: Timeout controlled by Rx FIFO 1	RW	0x0
0	ETOC	Enable Timeout Counter 0x0: Timeout Counter disabled 0x1: Timeout Counter enabled	RW	0x0

Table 12-38. MCAN_TOCV

Address Offset	0x0000 1A2C
Description	Timeout Counter Value Read/reset timeout counter.

Table 12-38. MCAN_TOCV (continued)

Type	RW

31	30	29	28	27	26	25	24	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																TOC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:0	TOC	Timeout Counter The Timeout Counter is decremented in multiples of CAN bit times [1-16] depending on the configuration of the MCAN_TSCC[19:16] TCP field. When decremented to zero, interrupt flag MCAN_IR[18] TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via the MCAN_TOCC[2:1] TOS field.	RWTC	0xFFFF

Table 12-39. MCAN_ECR

Address Offset	0x0000 1A40
Description	Error Counter Register State of Rx/Tx Error Counter, CAN Error Logging.
Type	R

31	30	29	28	27	26	25	24	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								CEL								RP	REC							TEC							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:16	CEL	CAN Error Logging The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to the MCAN_ECR[23:16] CEL field. The counter stops at 0xFF; the next increment of the MCAN_ECR[7:0] TEC or MCAN_ECR[14:8] REC fields sets interrupt flag MCAN_IR[22] ELO.	R	0x0
15	RP	Receive Error Passive 0x0: The Receive Error Counter is below the error passive level of 128 0x1: The Receive Error Counter has reached the error passive level of 128	R	0x0
14:8	REC	Receive Error Counter Actual state of the Receive Error Counter, values between 0 and 127.	R	0x0
7:0	TEC	Transmit Error Counter Actual state of the Transmit Error Counter, values between 0 and 255. **Note:** When the MCAN_CCCR[2] ASM bit is set, the CAN protocol controller does not increment the MCAN_ECR[7:0] TEC and MCAN_ECR[14:8] REC fields when a CAN protocol error is detected, but the MCAN_ECR[23:16] CEL field is still incremented.	R	0x0

Table 12-40. MCAN_PSR

Address Offset	0x0000 1A44
Description	Protocol Status Register CAN protocol controller status, transmitter delay compensation value.
Type	R

31	30	29	28	27	26	25	24	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								TDCV								RESERVED	PXE	RFDF	RBRS	RESI	DLEC			BO	EW	EP	ACT		LEC		

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R	0x0
22:16	TDCV	Transmitter Delay Compensation Value Position of the secondary sample point, defined by the sum of the measured delay from the MCAN_TX to MCAN_RX pins and the MCAN_TDCR[14:8] TDCO field. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq (0x00-0x7F).	R	0x0
15	RESERVED	Reserved	R	0x0
14	PXE	Protocol Exception Event 0x0: No protocol exception event occurred since last read access 0x1: Protocol exception event occurred	R	0x0
13	RFDF	Received a CAN FD Message This bit is set independent of acceptance filtering. 0x0: Since this bit was reset by the Host CPU, no CAN FD message has been received 0x1: Message in CAN FD format with FDF flag set has been received	R	0x0
12	RBRS	BRS flag of last received CAN FD Message This bit is set together with the MCAN_PSR[13] RFDF bit, independent of acceptance filtering. 0x0: Last received CAN FD message did not have its BRS flag set 0x1: Last received CAN FD message had its BRS flag set	R	0x0
11	RESI	ESI flag of last received CAN FD Message This bit is set together with the MCAN_PSR[13] RFDF bit, independent of acceptance filtering. 0x0: Last received CAN FD message did not have its ESI flag set 0x1: Last received CAN FD message had its ESI flag set	R	0x0
10:8	DLEC	Data Phase Last Error Code Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for the MCAN_PSR[2:0] LEC field. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.	R	0x7

Bits	Field Name	Description	Type	Reset
7	BO	Bus_Off Status 0x0: The MCAN module is not Bus_Off 0x1: The MCAN module is in Bus_Off state	R	0x0
6	EW	Warning Status 0x0: Both error counters are below the Error_Warning limit of 96 0x1: At least one of error counter has reached the Error_Warning limit of 96	R	0x0
5	EP	Error Passive 0x0: The MCAN module is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected 0x1: The MCAN module is in the Error_Passive state	R	0x0
4:3	ACT	Activity Monitors the module's CAN communication state. 0x0: Synchronizing - node is synchronizing on CAN communication 0x1: Idle - node is neither receiver nor transmitter 0x2: Receiver - node is operating as receiver 0x3: Transmitter - node is operating as transmitter **Note:** ACT is set to 00 by a Protocol Exception Event.	R	0x0

Bits	Field Name	Description	Type	Reset
2:0	LEC	Last Error Code The MCAN_PSR[2:0] LEC field indicates the type of the last error to occur on the CAN bus. This field will be cleared to 0 when a message has been transferred (reception or transmission) without error. 0x0: No Error: No error occurred since the MCAN_PSR[2:0] LEC field has been reset by successful reception or transmission. 0x1: Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 0x2: Form Error: A fixed format part of a received frame has the wrong format. 0x3: AckError: The message transmitted by the MCAN module was not acknowledged by another node. 0x4: Bit1Error: 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. 0x5: Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value 0), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the Host CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 0x6: CRCError: The CRC check sum of a received message was incorrect. The CRC of an incom-ing message does not match with the CRC calculated from the received data. 0x7: NoChange: Any read access to the Protocol Status Register re-initializes the MCAN_PSR[2:0] LEC field to '0x7'. When the MCAN_PSR[2:0] LEC field shows the value '0x7', no CAN bus event was detected since the last Host CPU read access to the Protocol Status Register. **Note:** When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in the MCAN_PSR[10:8] DLEC field instead of the MCAN_PSR[2:0] LEC field. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error. **Note:** The Bus_Off recovery sequence (see ISO11898-1:2015) cannot be shortened by setting or resetting the MCAN_CCCR[0] INIT bit. If the device goes Bus_Off, it will set the MCAN_CCCR[0] INIT bit of its own accord, stopping all bus activities. Once the MCAN_CCCR[0] INIT bit has been cleared by the Host CPU, the device 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 Management Counters will	R	0x7

Bits	Field Name	Description	Type	Reset
		be reset. During the waiting time after the resetting of the MCAN_CCCR[0] INIT bit, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to the MCAN_PSR[2:0] LEC field, enabling the Host CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. The MCAN_ECR[14:8] REC field is used to count these sequences.		

Table 12-41. MCAN_TDCR

Address Offset 0x0000 1A48

Description Transmitter Delay Comensation Register
Configuration of transmitter delay compensation offset and filter window length.

Type RW

31	30	29	28	27	26	25	24	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																TDCO								RESERVED						TDCF							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x0
14:8	TDCO	Transmitter Delay Compensation Offset Offset value defining the distance between the measured delay from the MCAN_RX and MCAN_TX pins and the secondary sample point. Valid values are 0 to 127 mtq (0x00-0x7F).	RW	0x0
7	RESERVED	Reserved	R	0x0
6:0	TDCF	Transmitter Delay Compensation Filter Window Length Defines the minimum value for the SSP position, dominant edges on the MCAN_RX pin that would result in an earlier SSP position are ignored for transmitter delay measure-ment. The feature is enabled when the MCAN_TDCR[6:0] TDCF field is configured to a value greater than the MCAN_TDCR[14:8] TDCO filed. Valid values are 0 to 127 mtq (0x00-0x7F).	RW	0x0

Table 12-42. MCAN_IR

Address Offset 0x0000 1A50

Description Interrupt Register
The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the Host CPU clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register. The configuration of the MCAN_IE register controls whether an interrupt is generated. The configuration of the MCAN_ILS register controls on which interrupt line an interrupt is signalled.

Type RW

31	30	29	28	27	26	25	24	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		ARA	PED	PEA	WDI	BO	EW	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29	ARA	Access to Reserved Address 0x0: No access to reserved address occurred 0x1: Access to reserved address occurred	RW1TC	0x0
28	PED	Protocol Error in Data Phase 0x0: No protocol error in data phase 0x1: Protocol error in data phase detected (MCAN_PSR[10:8] DLEC ≠ 0.7)	RW1TC	0x0
27	PEA	Protocol Error in Arbitration Phase 0x0: No protocol error in arbitration phase 0x1: Protocol error in arbitration phase detected (MCAN_PSR[2:0] LEC ≠ 0.7)	RW1TC	0x0
26	WDI	Watchdog Interrupt 0x0: No Message RAM Watchdog event occurred 0x1: Message RAM Watchdog event due to missing READY	RW1TC	0x0
25	BO	Bus_Off Status 0x0: Bus_Off status unchanged 0x1: Bus_Off status changed	RW1TC	0x0
24	EW	Warning Status 0x0: Error_Warning status unchanged 0x1: Error_Warning status changed	RW1TC	0x0
23	EP	Error Passive 0x0: Error_Passive status unchanged 0x1: Error_Passive status changed	RW1TC	0x0
22	ELO	Error Logging Overflow 0x0: CAN Error Logging Counter did not overflow 0x1: Overflow of CAN Error Logging Counter occurred	RW1TC	0x0
21	BEU	Bit Error Uncorrected Message RAM bit error detected, uncorrected. Controlled by input signal generated by parity/ECC logic attached to the Message RAM. An uncorrected Message RAM bit error sets the MCAN_CCCR[0] INIT bit to 1. This is done to avoid transmission of corrupted data. 0x0: No bit error detected when reading from Message RAM 0x1: Bit error detected, uncorrected (example: parity logic)	RW1TC	0x0
20	BEC	Bit Error Corrected Message RAM bit error detected and corrected. Controlled by input signal generated by parity/ECC logic attached to the Message RAM. 0x0: No bit error detected when reading from Message RAM 0x1: Bit error detected and corrected (example: ECC)	RW1TC	0x0
19	DRX	Message stored to Dedicated Rx Buffer The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0x0: No Rx Buffer updated 0x1: At least one received message stored into an Rx Buffer	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
18	TOO	Timeout Occurred 0x0: No timeout 0x1: Timeout reached	RW1TC	0x0
17	MRAF	Message RAM Access Failure The flag is set, when the Rx Handler: • has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. • was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated respectively the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN module is switched into Restricted Operation Mode (see Section 12.4.4.5). To leave Restricted Operation Mode, the Host CPU has to reset the MCAN_CCCR[2] ASM bit. 0x0: No Message RAM access failure occurred 0x1: Message RAM access failure occurred	RW1TC	0x0
16	TSW	Timestamp Wraparound 0x0: No timestamp counter wrap-around 0x1: Timestamp counter wrapped around	RW1TC	0x0
15	TEFL	Tx Event FIFO Element Lost 0x0: No Tx Event FIFO element lost 0x1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero	RW1TC	0x0
14	TEFF	Tx Event FIFO Full 0x0: Tx Event FIFO not full 0x1: Tx Event FIFO full	RW1TC	0x0
13	TEFW	Tx Event FIFO Watermark Reached 0x0: Tx Event FIFO fill level below watermark 0x1: Tx Event FIFO fill level reached watermark	RW1TC	0x0
12	TEFN	Tx Event FIFO New Entry 0x0: Tx Event FIFO unchanged 0x1: Tx Handler wrote Tx Event FIFO element	RW1TC	0x0
11	TFE	Tx FIFO Empty 0x0: Tx FIFO non-empty 0x1: Tx FIFO empty	RW1TC	0x0
10	TCF	Transmission Cancellation Finished 0x0: No transmission cancellation finished 0x1: Transmission cancellation finished	RW1TC	0x0
9	TC	Transmission Completed 0x0: No transmission completed 0x1: Transmission completed	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
8	HPM	High Priority Message 0x0: No high priority message received 0x1: High priority message received	RW1TC	0x0
7	RF1L	Rx FIFO 1 Message Lost 0x0: No Rx FIFO 1 message lost 0x1: Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero	RW1TC	0x0
6	RF1F	Rx FIFO 1 Full 0x0: Rx FIFO 1 not full 0x1: Rx FIFO 1 full	RW1TC	0x0
5	RF1W	Rx FIFO 1 Watermark Reached 0x0: Rx FIFO 1 fill level below watermark 0x1: Rx FIFO 1 fill level reached watermark	RW1TC	0x0
4	RF1N	Rx FIFO 1 New Message 0x0: No new message written to Rx FIFO 1 0x1: New message written to Rx FIFO 1	RW1TC	0x0
3	RF0L	Rx FIFO 0 Message Lost 0x0: No Rx FIFO 0 message lost 0x1: Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero	RW1TC	0x0
2	RF0F	Rx FIFO 0 Full 0x0: Rx FIFO 0 not full 0x1: Rx FIFO 0 full	RW1TC	0x0
1	RF0W	Rx FIFO 0 Watermark Reached 0x0: Rx FIFO 0 fill level below watermark 0x1: Rx FIFO 0 fill level reached watermark	RW1TC	0x0
0	RF0N	Rx FIFO 0 New Message 0x0: No new message written to Rx FIFO 0 0x1: New message written to Rx FIFO 0	RW1TC	0x0

Table 12-43. MCAN_IE

Address Offset 0x0000 1A54

Description Interrupt Enable
The settings in the Interrupt Enable register determine which status changes in the Interrupt Register are signalled on an interrupt line.

Type RW

31	30	29	28	27	26	25	24	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		AREE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	BEUE	BECE	DRX	TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29	ARAE	Access to Reserved Address Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
28	PEDE	Protocol Error in Data Phase Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
27	PEAE	Protocol Error in Arbitration Phase Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
26	WDIE	Watchdog Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
25	BOE	Bus_Off Status Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
24	EWE	Warning Status Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
23	EPE	Error Passive Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
22	ELOE	Error Logging Overflow Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
21	BEUE	Bit Error Uncorrected Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
20	BECE	Bit Error Corrected Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
19	DRX	Message stored to Dedicated Rx Buffer Interrupt Enable	RW	0x0
18	TOOE	Timeout Occurred Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
17	MRAFE	Message RAM Access Failure Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
16	TSWE	Timestamp Wraparound Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
15	TEFLE	Tx Event FIFO Event Lost Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
14	TEFFE	Tx Event FIFO Full Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
13	TEFWE	Tx Event FIFO Watermark Reached Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
12	TEFNE	Tx Event FIFO New Entry Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
11	TFEE	Tx FIFO Empty Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
10	TCFE	Transmission Cancellation Finished Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
9	TCE	Transmission Completed Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
8	HPME	High Priority Message Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
7	RF1LE	Rx FIFO 1 Message Lost Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
6	RF1FE	Rx FIFO 1 Full Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
5	RF1WE	Rx FIFO 1 Watermark Reached Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
4	RF1NE	Rx FIFO 1 New Message Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
3	RF0LE	Rx FIFO 0 Message Lost Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
2	RF0FE	Rx FIFO 0 Full Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
1	RF0WE	Rx FIFO 0 Watermark Reached Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
0	RF0NE	Rx FIFO 0 New Message Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0

Table 12-44. MCAN_ILS

Address Offset	0x0000 1A58
Description	Interrupt Line Select The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines. For interrupt generation the respective interrupt line has to be enabled via the MCAN_ILE[0] EINT0 and MCAN_ILE[1] EINT1 bits.
Type	RW

31	30	29	28	27	26	25	24	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		ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
29	ARAL	Access to Reserved Address Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
28	PEDL	Protocol Error in Data Phase Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
27	PEAL	Protocol Error in Arbitration Phase Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
26	WDIL	Watchdog Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
25	BOL	Bus_Off Status Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
24	EWL	Warning Status Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
23	EPL	Error Passive Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
22	ELOL	Error Logging Overflow Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
21	BEUL	Bit Error Uncorrected Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
20	BECL	Bit Error Corrected Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
19	DRXL	Message stored to Dedicated Rx Buffer Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
18	TOOL	Timeout Occurred Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
17	MRAFL	Message RAM Access Failure Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
16	TSWL	Timestamp Wraparound Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
15	TEFLL	Tx Event FIFO Event Lost Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
14	TEFFL	Tx Event FIFO Full Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0

Bits	Field Name	Description	Type	Reset
13	TEFWL	Tx Event FIFO Watermark Reached Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
12	TEFNL	Tx Event FIFO New Entry Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
11	TFEL	Tx FIFO Empty Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
10	TCFL	Transmission Cancellation Finished Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
9	TCL	Transmission Completed Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
8	HPML	High Priority Message Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
7	RF1LL	Rx FIFO 1 Message Lost Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
6	RF1FL	Rx FIFO 1 Full Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
5	RF1WL	Rx FIFO 1 Watermark Reached Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
4	RF1NL	Rx FIFO 1 New Message Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
3	RF0LL	Rx FIFO 0 Message Lost Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
2	RF0FL	Rx FIFO 0 Full Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
1	RF0WL	Rx FIFO 0 Watermark Reached Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
0	RF0NL	Rx FIFO 0 New Message Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0

Table 12-45. MCAN_ILE

Address Offset	0x0000 1A5C
Description	Interrupt Line Enable Enable/disable interrupt lines INT0/INT1. Each of the two interrupt lines to the Host CPU can be enabled/disabled separately by programming the MCAN_ILE[0] EINT0 and MCAN_ILE[1] EINT1 bits.

Table 12-45. MCAN_ILE (continued)

Type RW

31	30	29	28	27	26	25	24	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																														EINT1	EINT0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	EINT1	Enable Interrupt Line 1 0x0: Interrupt line INT1 disabled 0x1: Interrupt line INT1 enabled	RW	0x0
0	EINT0	Enable Interrupt Line 0 0x0: Interrupt line INT0 disabled 0x1: Interrupt line INT0 enabled	RW	0x0

Table 12-46. MCAN_GFC

Address Offset 0x0000 1A80

Description Global Filter Configuration
Handling of non-matching frames and remote frames.
Global settings for Message ID filtering. The Global Filter Configuration controls the filter path for standard and extended messages (see Figure 12-8 and Figure 12-9).

Type RW

31	30	29	28	27	26	25	24	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																										ANFS		ANFE		RRFS	RRFE

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:4	ANFS	Accept Non-matching Frames Standard Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 0x0: Accept in Rx FIFO 0 0x1: Accept in Rx FIFO 1 0x2: Reject 0x3: Reject	RW	0x0
3:2	ANFE	Accept Non-matching Frames Extended Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 0x0: Accept in Rx FIFO 0 0x1: Accept in Rx FIFO 1 0x2: Reject 0x3: Reject	RW	0x0
1	RRFS	Reject Remote Frames Standard 0x0: Filter remote frames with 11-bit standard IDs 0x1: Reject all remote frames with 11-bit standard IDs	RW	0x0
0	RRFE	Reject Remote Frames Extended 0x0: Filter remote frames with 29-bit extended IDs 0x1: Reject all remote frames with 29-bit extended IDs	RW	0x0

Table 12-47. MCAN_SIDFC

Address Offset 0x0000 1A84

Table 12-47. MCAN_SIDFC (continued)

Description	Standard ID Filter Configuration Number of filter elements, pointer to start of filter list. Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages (see Figure 12-8).
Type	RW

31	30	29	28	27	26	25	24	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								LSS								FLSSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:16	LSS	List Size Standard 0x0: No standard Message ID filter 0x1-0x80 (1-128): Number of standard Message ID filter elements > 0x80 (128): Values greater than 128 are interpreted as 128	RW	0x0
15:2	FLSSA	Filter List Standard Start Address Start address of standard Message ID filter list (32-bit word address, see Figure 12-14).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 12-48. MCAN_XIDFC

Address Offset	0x0000 1A88
Description	Extended ID Filter Configuration Number of filter elements, pointer to start of filter list. Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for standard messages (see Figure 12-9).
Type	RW

31	30	29	28	27	26	25	24	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								LSE								FLESA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R	0x0
22:16	LSE	List Size Extended 0x0: No extended Message ID filter 0x1-0x40 (1-64): Number of extended Message ID filter elements > 0x40 (64): Values greater than 64 are interpreted as 64	RW	0x0
15:2	FLESA	Filter List Extended Start Address Start address of extended Message ID filter list (32-bit word address, see Figure 12-14).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 12-49. MCAN_XIDAM

Address Offset	0x0000 1A90
Description	Extended ID AND Mask 29-bit logical AND mask for J1939.
Type	RW

31	30	29	28	27	26	25	24	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	EIDM

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:0	EIDM	Extended ID Mask For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.	RW	0x1FFFFFFF

Table 12-50. MCAN_HPMS

Address Offset 0x0000 1A94

Description High Priority Message Status
Status monitoring of incoming high priority messages.
This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Type R

31	30	29	28	27	26	25	24	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																FLST	FIDX							MSI		BIDX					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15	FLST	Filter List Indicates the filter list of the matching filter element. 0x0: Standard Filter List 0x1: Extended Filter List	R	0x0
14:8	FIDX	Filter Index Index of matching filter element. Range is 0 to MCAN_SIDFC[23:16] LSS - 1 respectively MCAN_XIDFC[22:16] LSE - 1.	R	0x0
7:6	MSI	Message Storage Indicator 0x0: No FIFO selected 0x1: FIFO message lost 0x2: Message stored in FIFO 0 0x3: Message stored in FIFO 1	R	0x0
5:0	BIDX	Buffer Index Index of Rx FIFO element to which the message was stored. Only valid when the MCAN_HPMS[7:6] MSI = 1.	R	0x0

Table 12-51. MCAN_NDAT1

Address Offset 0x0000 1A98

Description New Data 1
NewDat flags of dedicated Rx buffers 0-31.
The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host CPU clears them. Aflag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. Ahard reset will clear the register.

Type RW

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

ADVANCE INFORMATION

ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0

Bits	Field Name	Description	Type	Reset
31	ND31	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
30	ND30	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
29	ND29	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
28	ND28	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
27	ND27	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
26	ND26	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
25	ND25	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
24	ND24	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
23	ND23	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
22	ND22	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
21	ND21	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
20	ND20	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
19	ND19	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
18	ND18	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
17	ND17	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
16	ND16	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
15	ND15	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
14	ND14	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
13	ND13	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
12	ND12	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
11	ND11	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
10	ND10	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
9	ND9	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
8	ND8	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
7	ND7	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
6	ND6	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
5	ND5	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
4	ND4	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
3	ND3	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
2	ND2	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
1	ND1	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
0	ND0	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Table 12-52. MCAN_NDAT2

Address Offset	0x0000 1A9C
Description	New Data 2 NewDat flags of dedicated Rx buffers 32-63. The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host CPU clears them. Aflag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. Ahard reset will clear the register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32

Bits	Field Name	Description	Type	Reset
31	ND63	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
30	ND62	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
29	ND61	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
28	ND60	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
27	ND59	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
26	ND58	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
25	ND57	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
24	ND56	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
23	ND55	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
22	ND54	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
21	ND53	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
20	ND52	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
19	ND51	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
18	ND50	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
17	ND49	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
16	ND48	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
15	ND47	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
14	ND46	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
13	ND45	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
12	ND44	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
11	ND43	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
10	ND42	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
9	ND41	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
8	ND40	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
7	ND39	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
6	ND38	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
5	ND37	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
4	ND36	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
3	ND35	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
2	ND34	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
1	ND33	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
0	ND32	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Table 12-53. MCAN_RXF0C

Address Offset	0x0000 1AA0
Description	Rx FIFO 0 Configuration FIFO 0 operation mode, watermark, size and start address.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0OM					F0WM			RESERVED				F0S				F0SA														RESERVED		

Bits	Field Name	Description	Type	Reset
31	F0OM	FIFO 0 Operation Mode FIFO 0 can be operated in blocking or in overwrite mode (see Section 12.4.8.2). 0x0: FIFO 0 blocking mode 0x1: FIFO 0 overwrite mode	RW	0x0
30:24	F0WM	Rx FIFO 0 Watermark 0x0: Watermark interrupt disabled 0x1-0x40 (1-64): Level for Rx FIFO 0 watermark interrupt (MCAN_IR[1] RF0W) > 0x40 (64): Watermark interrupt disabled	RW	0x0
23	RESERVED	Reserved	R	0x0
22:16	F0S	Rx FIFO 0 Size 0x0: No Rx FIFO 0 0x1-0x40 (1-64): Number of Rx FIFO 0 elements > 0x40 (64): Values greater than 64 are interpreted as 64 The Rx FIFO 0 elements are indexed from 0 to MCAN_RXF0C[22:16] F0S - 1	RW	0x0
15:2	F0SA	Rx FIFO 0 Start Address Start address of Rx FIFO 0 in Message RAM (32-bit word address, see Figure 12-14).	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	RESERVED	Reserved	R	0x0

Table 12-54. MCAN_RXF0S

Address Offset 0x0000 1AA4

Description Rx FIFO 0 Status
FIFO 0 message lost/full indication, put index, get index and fill level.

Type R

31	30	29	28	27	26	25	24	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						RF0L	F0F	RESERVED								F0PI								RESERVED							
																								F0GI							
RESERVED								F0FL																							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	RF0L	Rx FIFO 0 Message Lost This bit is a copy of interrupt flag MCAN_IR[3] RF0L. When the MCAN_IR[3] RF0L flag is reset, this bit is also reset. 0x0: No Rx FIFO 0 message lost 0x1: Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero **Note:** Overwriting the oldest message when the MCAN_RXF0C[31] F0OM = 1 will not set this flag.	R	0x0
24	F0F	Rx FIFO 0 Full 0x0: Rx FIFO 0 not full 0x1: Rx FIFO 0 full	R	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	F0PI	Rx FIFO 0 Put Index Rx FIFO 0 write index pointer, range 0 to 63.	R	0x0
15:14	RESERVED	Reserved	R	0x0
13:8	F0GI	Rx FIFO 0 Get Index Rx FIFO 0 read index pointer, range 0 to 63.	R	0x0
7	RESERVED	Reserved	R	0x0
6:0	F0FL	Rx FIFO 0 Fill Level Number of elements stored in Rx FIFO 0, range 0 to 64.	R	0x0

Table 12-55. MCAN_RXF0A

Address Offset 0x0000 1AA8

Description Rx FIFO 0 Acknowledge
FIFO 0 acknowledge last index of read buffers, updates get index and fill level.

Type RW

31	30	29	28	27	26	25	24	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																										F0AI					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
5:0	F0AI	Rx FIFO 0 Acknowledge Index After the Host CPU has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to the MCAN_RXF0A[5:0] F0AI field. This will set the Rx FIFO 0 Get Index MCAN_RXF0S[13:8] F0GI field to the MCAN_RXF0A[5:0] F0AI field + 1 and update the FIFO 0 Fill Level MCAN_RXF0S[6:0] F0FL field.	RW	0x0

Table 12-56. MCAN_RXBC

Address Offset	0x0000 1AAC
Description	Rx Buffer Configuration Start address of Rx buffer section.
Type	RW

31	30	29	28	27	26	25	24	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																RBSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:2	RBSA	Rx Buffer Start Address Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address, see Figure 12-14). Also used to reference debug messages A,B,C. **Note:** Debug feature is not supported.	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 12-57. MCAN_RXF1C

Address Offset	0x0000 1AB0
Description	Rx FIFO 1 Configuration FIFO 1 operation mode, watermark, size and start address.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1OM				F1WM				RESERVED				F1S				F1SA														RESERVED	

Bits	Field Name	Description	Type	Reset
31	F1OM	FIFO 1 Operation Mode FIFO 1 can be operated in blocking or in overwrite mode (see Section 12.4.8.2). 0x0: FIFO 1 blocking mode 0x1: FIFO 1 overwrite mode	RW	0x0
30:24	F1WM	Rx FIFO 1 Watermark 0x0: Watermark interrupt disabled 0x1-0x40 (1-64): Level for Rx FIFO 1 watermark interrupt (MCAN_IR[5] RF1W) > 0x40 (64): Watermark interrupt disabled	RW	0x0
23	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
22:16	F1S	Rx FIFO 1 Size 0x0: No Rx FIFO 1 0x1-0x40 (1-64): Number of Rx FIFO 1 elements > 0x40 (64): Values greater than 64 are interpreted as 64 The Rx FIFO 1 elements are indexed from 0 to MCAN_RXF1C[22:16] F1S - 1	RW	0x0
15:2	F1SA	Rx FIFO 1 Start Address Start address of Rx FIFO 1 in Message RAM (32-bit word address, see Figure 12-14).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 12-58. MCAN_RXF1S

Address Offset	0x0000 1AB4
Description	Rx FIFO 1 Status FIFO 1 message lost/full indication, put index, get index and fill level.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMS		RESERVED				RF1L	F1F	RESERVED		F1PI						RESERVED		F1GI						RESERVED		F1FL					

Bits	Field Name	Description	Type	Reset
31:30	DMS	Debug Message Status 0x0: Idle state, wait for reception of debug messages, DMA request is cleared 0x1: Debug message A received 0x2: Debug messages A, B received 0x3: Debug messages A, B, C received, DMA request is set **Note:** Debug feature is not supported.	R	0x0
29:26	RESERVED	Reserved	R	0x0
25	RF1L	Rx FIFO 1 Message Lost This bit is a copy of interrupt flag MCAN_IR[7] RF1L. When the MCAN_IR[7] RF1L flag is reset, this bit is also reset. 0x0: No Rx FIFO 1 message lost 0x1: Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero **Note:** Overwriting the oldest message when the MCAN_RXF1C[31] F1OM = 1 will not set this flag.	R	0x0
24	F1F	Rx FIFO 1 Full 0x0: Rx FIFO 1 not full 0x1: Rx FIFO 1 full	R	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	F1PI	Rx FIFO 1 Put Index Rx FIFO 1 write index pointer, range 0 to 63.	R	0x0
15:14	RESERVED	Reserved	R	0x0
13:8	F1GI	Rx FIFO 1 Get Index Rx FIFO 1 read index pointer, range 0 to 63.	R	0x0
7	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
6:0	F1FL	Rx FIFO 1 Fill Level Number of elements stored in Rx FIFO 1, range 0 to 64.	R	0x0

Table 12-59. MCAN_RXF1A

Address Offset	0x0000 1AB8
Description	Rx FIFO 1 Acknowledge FIFO 1 acknowledge last index of read buffers, updates get index and fill level.
Type	RW

31	30	29	28	27	26	25	24	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																								F1AI							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:0	F1AI	Rx FIFO 1 Acknowledge Index After the Host CPU has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to the MCAN_RXF1A[5:0] F1AI field. This will set the Rx FIFO 1 Get Index MCAN_RXF1S[13:8] F1GI field to the MCAN_RXF1A[5:0] F1AI field + 1 and update the FIFO 1 Fill Level MCAN_RXF1S[6:0] F1FL field.	RW	0x0

Table 12-60. MCAN_RXESC

Address Offset	0x0000 1ABC
Description	Rx Buffer/FIFO Element Size Configuration Configure data field size for storage of accepted frames.
Type	RW

31	30	29	28	27	26	25	24	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																					RBDS			RESERVED			F1DS			RESERVED			F0DS		

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x0
10:8	RBDS	Rx Buffer Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field	RW	0x0
7	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
6:4	F1DS	Rx FIFO 1 Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field	RW	0x0
3	RESERVED	Reserved	R	0x0
2:0	F0DS	Rx FIFO 0 Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field **Note:** In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by the MCAN_RXESC register are stored to the Rx Buffer respectively Rx FIFO element. The rest of the frame's data field is ignored.	RW	0x0

Table 12-61. MCAN_TXBC

Address Offset	0x0000 1AC0
Description	Tx Buffer Configuration Configure Tx FIFO/Queue mode, Tx FIFO/Queue size, number of dedicated Tx buffers, Tx buffer start address.
Type	RW

31	30	29	28	27	26	25	24	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	TFQM	TFQS						RESERVED		NDTB						TBSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0x0
30	TFQM	Tx FIFO/Queue Mode 0x0: Tx FIFO operation 0x1: Tx Queue operation	RW	0x0
29:24	TFQS	Transmit FIFO/Queue Size 0x0: No Tx FIFO/Queue 0x1-0x20 (1-32): Number of Tx Buffers used for Tx FIFO/ Queue > 0x20 (32): Values greater than 32 are interpreted as 32	RW	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	NDTB	Number of Dedicated Transmit Buffers 0x0: No Dedicated Tx Buffers 0x1-0x20 (1-32): Number of Dedicated Tx Buffers > 0x20 (32): Values greater than 32 are interpreted as 32	RW	0x0

Bits	Field Name	Description	Type	Reset
15:2	TBSA	Tx Buffers Start Address Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 12-14). **Note:** Be aware that the sum of the MCAN_TXBC[29:24] TFQS and MCAN_TXBC[21:16] NDTB fields may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 12-62. MCAN_TXFQS

Address Offset	0x0000 1AC4
Description	Tx FIFO/Queue Status Tx FIFO/Queue full indication and put index, Tx FIFO get index and fill level. The Tx FIFO/Queue status is related to the pending Tx requests listed in the MCAN_TXBRP register. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (the MCAN_TXBRP register not yet updated).
Type	R

31	30	29	28	27	26	25	24	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										TFQF	TFQPI					RESERVED			TFGI					RESERVED		TFFL					

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x0
21	TFQF	Tx FIFO/Queue Full 0x0: Tx FIFO/Queue not full 0x1: Tx FIFO/Queue full	R	0x0
20:16	TFQPI	Tx FIFO/Queue Put Index Tx FIFO/Queue write index pointer, range 0 to 31.	R	0x0
15:13	RESERVED	Reserved	R	0x0
12:8	TFGI	Tx FIFO Get Index Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (MCAN_TXBC[30] TFQM = 1).	R	0x0
7:6	RESERVED	Reserved	R	0x0
5:0	TFFL	Tx FIFO Free Level Number of consecutive free Tx FIFO elements starting from the MCAN_TXFQS[12:8] TFGI field, range 0 to 32. Read as zero when Tx Queue operation is configured (MCAN_TXBC[30] TFQM = 1) **Note:** In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. **Example:** For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.	R	0x0

Table 12-63. MCAN_TXESC

Address Offset	0x0000 1AC8

Table 12-63. MCAN_TXESC (continued)

Description	Tx Buffer Element Size Configuration Configure data field size for frame transmission. Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.
Type	RW

31	30	29	28	27	26	25	24	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																													TBDS		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0
2:0	TBDS	Tx Buffer Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field **Note:** In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size MCAN_TXESC[2:0] TBDS, the bytes not defined by the Tx Buffer are transmitted as '0xCC' (padding bytes).	RW	0x0

Table 12-64. MCAN_TXBRP

Address Offset	0x0000 1ACC
Description	Tx Buffer Request Pending Tx buffers with pending transmission request. Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via the MCAN_TXBAR register. The bits are reset after a requested transmission has completed or has been cancelled via the MCAN_TXBCR register. The MCAN_TXBRP bits are set only for those Tx Buffers configured via the MCAN_TXBC register. After a MCAN_TXBRP bit has been set, a Tx scan (see Section 12.4.9, *Tx Handling*) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID). A cancellation request resets the corresponding transmission request pending bit of register the MCAN_TXBRP register. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding MCAN_TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signalled via the MCAN_TXBCF • after successful transmission together with the corresponding MCAN_TXBTO bit • when the transmission has not yet been started at the point of cancellation • when the transmission has been aborted due to lost arbitration • when an error occurred during frame transmission In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding MCAN_TXBCF bit is set for all unsuccessful transmissions. **Note:** The TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding MCAN_TXBRP bit is reset.
Type	R

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

ADVANCE INFORMATION

TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0

Bits	Field Name	Description	Type	Reset
31	TRP31	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
30	TRP30	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
29	TRP29	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
28	TRP28	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
27	TRP27	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
26	TRP26	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
25	TRP25	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
24	TRP24	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
23	TRP23	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
22	TRP22	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
21	TRP21	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
20	TRP20	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
19	TRP19	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
18	TRP18	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
17	TRP17	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0

Bits	Field Name	Description	Type	Reset
16	TRP16	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
15	TRP15	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
14	TRP14	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
13	TRP13	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
12	TRP12	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
11	TRP11	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
10	TRP10	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
9	TRP9	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
8	TRP8	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
7	TRP7	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
6	TRP6	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
5	TRP5	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
4	TRP4	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
3	TRP3	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
2	TRP2	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
1	TRP1	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0

Bits	Field Name	Description	Type	Reset
0	TRP0	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0

Table 12-65. MCAN_TXBAR

Address Offset	0x0000 1AD0
Description	Tx Buffer Add Request Add transmission requests. Each Tx Buffer has its own Add Request bit. Writing a '1' will set the corresponding Add Request bit; writing a '0' has no impact. This enables the Host CPU to set transmission requests for multiple Tx Buffers with one write to the MCAN_TXBAR register. The MCAN_TXBAR bits are set only for those Tx Buffers configured via the MCAN_TXBC register. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed. **Note:** If an add request is applied for a Tx Buffer with pending transmission request (corresponding MCAN_TXBRP bit already set), this add request is ignored.
Type	RW

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

Bits	Field Name	Description	Type	Reset
31	AR31	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
30	AR30	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
29	AR29	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
28	AR28	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
27	AR27	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
26	AR26	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
25	AR25	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
24	AR24	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
23	AR23	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
22	AR22	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
21	AR21	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
20	AR20	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
19	AR19	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
18	AR18	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
17	AR17	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
16	AR16	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
15	AR15	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
14	AR14	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
13	AR13	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
12	AR12	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
11	AR11	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
10	AR10	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
9	AR9	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
8	AR8	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
7	AR7	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
6	AR6	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
5	AR5	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
4	AR4	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
3	AR3	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
2	AR2	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
1	AR1	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
0	AR0	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0

Table 12-66. MCAN_TXBCR

Address Offset 0x0000 1AD4

Description Tx Buffer Cancellation Request
Request cancellation of pending transmissions.
Each Tx Buffer has its own Cancellation Request bit. Writing a '1' will set the corresponding Cancellation Request bit; writing a '0' has no impact. This enables the Host CPU to set cancellation requests for multiple Tx Buffers with one write to the MCAN_TXBCR register. The MCAN_TXBCR bits are set only for those Tx Buffers configured via the MCAN_TXBC register. The bits remain set until the corresponding bit of the MCAN_TXBRP register is reset.

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

Bits	Field Name	Description	Type	Reset
31	CR31	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
30	CR30	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
29	CR29	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
28	CR28	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
27	CR27	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
26	CR26	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
25	CR25	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
24	CR24	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
23	CR23	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
22	CR22	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
21	CR21	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
20	CR20	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
19	CR19	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
18	CR18	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
17	CR17	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
16	CR16	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
15	CR15	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
14	CR14	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
13	CR13	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
12	CR12	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
11	CR11	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
10	CR10	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
9	CR9	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
8	CR8	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
7	CR7	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
6	CR6	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
5	CR5	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
4	CR4	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
3	CR3	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
2	CR2	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
1	CR1	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
0	CR0	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0

Table 12-67. MCAN_TXBTO

Address Offset	0x0000 1AD8
Description	Tx Buffer Transmission Occurred Signals successful transmissions, set when corresponding MCAN_TXBRP flag is cleared. Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register the MCAN_TXBAR register.
Type	R

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

Bits	Field Name	Description	Type	Reset
31	TO31	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
30	TO30	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
29	TO29	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
28	TO28	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
27	TO27	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
26	TO26	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
25	TO25	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
24	TO24	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
23	TO23	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
22	TO22	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
21	TO21	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
20	TO20	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
19	TO19	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
18	TO18	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
17	TO17	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
16	TO16	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0

Bits	Field Name	Description	Type	Reset
15	TO15	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
14	TO14	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
13	TO13	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
12	TO12	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
11	TO11	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
10	TO10	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
9	TO9	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
8	TO8	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
7	TO7	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
6	TO6	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
5	TO5	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
4	TO4	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
3	TO3	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
2	TO2	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
1	TO1	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
0	TO0	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0

Table 12-68. MCAN_TXBCF

Address Offset	0x0000 1ADC
Description	Tx Buffer Cancellation Finished Signals successful transmit cancellation, set when corresponding TXBRP flag is cleared after cancellation request. Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a cancellation was requested via the MCAN_TXBCR register. In case the corresponding MCAN_TXBRP bit was not set at the point of cancellation, MCAN_TXBCF[n] CF bit is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of the MCAN_TXBAR register.
Type	R

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

Bits	Field Name	Description	Type	Reset
31	CF31	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
30	CF30	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
29	CF29	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
28	CF28	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
27	CF27	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
26	CF26	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
25	CF25	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
24	CF24	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
23	CF23	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
22	CF22	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
21	CF21	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0

Bits	Field Name	Description	Type	Reset
20	CF20	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
19	CF19	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
18	CF18	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
17	CF17	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
16	CF16	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
15	CF15	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
14	CF14	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
13	CF13	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
12	CF12	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
11	CF11	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
10	CF10	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
9	CF9	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
8	CF8	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
7	CF7	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
6	CF6	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
5	CF5	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0

Bits	Field Name	Description	Type	Reset
4	CF4	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
3	CF3	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
2	CF2	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
1	CF1	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
0	CF0	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0

Table 12-69. MCAN_TXBTIE

Address Offset	0x0000 1AE0
Description	Tx Buffer Transmission Interrupt Enable Enable transmit interrupts for selected Tx buffers.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0

Bits	Field Name	Description	Type	Reset
31	TIE31	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
30	TIE30	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
29	TIE29	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
28	TIE28	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
27	TIE27	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
26	TIE26	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
25	TIE25	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0

Bits	Field Name	Description	Type	Reset
24	TIE24	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
23	TIE23	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
22	TIE22	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
21	TIE21	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
20	TIE20	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
19	TIE19	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
18	TIE18	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
17	TIE17	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
16	TIE16	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
15	TIE15	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
14	TIE14	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
13	TIE13	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
12	TIE12	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
11	TIE11	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
10	TIE10	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
9	TIE9	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0

Bits	Field Name	Description	Type	Reset
8	TIE8	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
7	TIE7	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
6	TIE6	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
5	TIE5	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
4	TIE4	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
3	TIE3	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
2	TIE2	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
1	TIE1	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
0	TIE0	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0

Table 12-70. MCAN_TXBCIE

Address Offset	0x0000 1AE4
Description	Tx Buffer Cancellation Finished Interrupt Enable Enable cancellation finished interrupts for selected Tx buffers.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0

Bits	Field Name	Description	Type	Reset
31	CFIE31	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
30	CFIE30	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
29	CFIE29	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
28	CFIE28	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
27	CFIE27	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
26	CFIE26	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
25	CFIE25	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
24	CFIE24	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
23	CFIE23	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
22	CFIE22	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
21	CFIE21	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
20	CFIE20	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
19	CFIE19	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
18	CFIE18	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
17	CFIE17	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
16	CFIE16	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
15	CFIE15	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
14	CFIE14	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
13	CFIE13	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
12	CFIE12	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
11	CFIE11	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
10	CFIE10	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
9	CFIE9	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
8	CFIE8	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
7	CFIE7	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
6	CFIE6	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
5	CFIE5	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
4	CFIE4	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
3	CFIE3	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
2	CFIE2	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
1	CFIE1	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
0	CFIE0	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0

Table 12-71. MCAN_TXEFC

Address Offset	0x0000 1AF0
Description	Tx Event FIFO Configuration Tx event FIFO watermark, size and start address.
Type	RW

31	30	29	28	27	26	25	24	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				EFWM				RESERVED				EFS				EFSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29:24	EFWM	Event FIFO Watermark 0x0: Watermark interrupt disabled 0x1-0x20 (1-32): Level for Tx Event FIFO watermark interrupt (MCAN_IR[13] TEFW) > 0x20 (32): Watermark interrupt disabled	RW	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	EFS	Event FIFO Size 0x0: Tx Event FIFO disabled 0x1-0x20 (1-32): Number of Tx Event FIFO elements > 0x20 (32): Values greater than 32 are interpreted as 32 The Tx Event FIFO elements are indexed from 0 to MCAN_TXEFC[21:16] EFS field - 1	RW	0x0
15:2	EFSA	Event FIFO Start Address Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 12-14).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 12-72. MCAN_TXEFS

Address Offset 0x0000 1AF4

Description Tx Event FIFO Status
Tx event FIFO element lost/full indication, put index, get index, and fill level.

Type R

31	30	29	28	27	26	25	24	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						TEFL	EFF	RESERVED				EFPI				RESERVED				EFGI				RESERVED		EFFL					

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	TEFL	This bit is a copy of interrupt flag MCAN_IR[15] TEFL. When the MCAN_IR[15] TEFL flag is reset, this bit is also reset. 0x0: No Tx Event FIFO element lost 0x1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.	R	0x0
24	EFF	Event FIFO Full 0x0: Tx Event FIFO not full 0x1: Tx Event FIFO full	R	0x0
23:21	RESERVED	Reserved	R	0x0
20:16	EFPI	Event FIFO Put Index Tx Event FIFO write index pointer, range 0 to 31.	R	0x0
15:13	RESERVED	Reserved	R	0x0
12:8	EFGI	Event FIFO Get Index Tx Event FIFO read index pointer, range 0 to 31.	R	0x0
7:6	RESERVED	Reserved	R	0x0
5:0	EFFL	Event FIFO Fill Level Number of elements stored in Tx Event FIFO, range 0 to 32.	R	0x0

Table 12-73. MCAN_TXEFA

Address Offset 0x0000 1AF8

Table 12-73. MCAN_TXEFA (continued)

Description Tx Event FIFO Acknowledge
Tx event FIFO acknowledge last index of read elements, updates get index and fill level.

Type RW

31	30	29	28	27	26	25	24	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																													EFAI		

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4:0	EFAI	After the Host CPU has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to the MCAN_TXEFA[4:0] EFAI field. This will set the Tx Event FIFO Get Index MCAN_TXEFS[12:8] EFGI field to the MCAN_TXEFA[4:0] EFAI field + 1 and update the Event FIFO Fill Level MCAN_TXEFS[5:0] EFFL field.	RW	0x0

Table 12-74. MCANSS_ECC_AGGR_REVISION

Address Offset 0x0000 1C00

Description Aggregator Revision Register
Revision parameters.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		BU		MODULE_ID												REVRTL					REVMAJ			CUSTOM				REVMIN			

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	Scheme	R	0x1
29:28	BU	Business Unit	R	0x2
27:16	MODULE_ID	Module ID	R	0x6A0
15:11	REVRTL	RTL version	R	0x1
10:8	REVMAJ	Major version	R	0x3
7:6	CUSTOM	Custom version	R	0x0
5:0	REVMIN	Minor version	R	0x0

Table 12-75. MCANSS_ECC_VECTOR

Address Offset 0x0000 1C08

Description ECC Vector Register
ECC Vector Register.

Type RW

31	30	29	28	27	26	25	24	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							RD_SVBUS_DONE	RD_SVBUS_ADDRESS								RD_SVBUS	RESERVED							ECC_VECTOR							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x0
24	RD_SVBUS_DONE	Status to indicate if read is complete	R	0x0

Bits	Field Name	Description	Type	Reset
23:16	RD_SVBUS_ADDRESS	Read address	RW	0x0
15	RD_SVBUS	Write 1 to trigger a read	RW	0x0
14:11	RESERVED	Reserved	R	0x0
10:0	ECC_VECTOR	Value written to select the corresponding ECC RAM for control or status	RW	0x0

Table 12-76. MCANSS_ECC_MISC_STATUS

Address Offset 0x0000 1C0C

Description Misc Status
Misc Status.

Type R

31	30	29	28	27	26	25	24	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																					NUM_RAMS										

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x0
10:0	NUM_RAMS	Indicates the number of RAMS serviced by the ECC aggregator	R	0x1

Table 12-77. MCANSS_ECC_WRAP_REVISION

Address Offset 0x0000 1C10

Description ECC Wrapper Revision Register
Revision parameters.

Type R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME				BU				MODULE_ID								REVRTL				REVMAJ				CUSTOM				REVMIN			

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	Scheme	R	0x1
29:28	BU	Business Unit	R	0x2
27:16	MODULE_ID	Module ID	R	0x6A4
15:11	REVRTL	RTL version	R	0x1
10:8	REVMAJ	Major version	R	0x1
7:6	CUSTOM	Custom version	R	0x0
5:0	REVMIN	Minor version	R	0x0

Table 12-78. MCANSS_ECC_CONTROL

Address Offset 0x0000 1C14

Description ECC Control
ECC Control Register.

Type RW

31	30	29	28	27	26	25	24	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																								ERR_ONCE	FORCE_NR_OW	FORCE_DED	FORCE_SEC	ENABLE_RMW	ECC_CHECK	ECC_ENABLE	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6	ERROR_ONCE	Force Error only once	RW	0x0
5	FORCE_N_ROW	Force Error on any RAM read	RW	0x0
4	FORCE_DED	Force Double Bit Error	RW	0x0
3	FORCE_SEC	Force Single Bit Error	RW	0x0
2	ENABLE_RMW	Enable RMW	RW	0x1
1	ECC_CHECK	Enable ECC check	RW	0x1
0	ECC_ENABLE	Enable ECC	RW	0x1

Table 12-79. MCANSS_ECC_ERR_CTRL1

Address Offset 0x0000 1C18

Description ECC Error Control1 Register
ECC Error Control1 Register.

Type RW

31	30	29	28	27	26	25	24	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															

Bits	Field Name	Description	Type	Reset
31:16	ECC_BIT1	Data bit that needs to be flipped when FORCE_SEC is set	RW	0x0
15:0	ECC_ROW	Row address where single or double-bit error needs to be applied. This is ignored if FORCE_N_ROW is set.	RW	0x0

Table 12-80. MCANSS_ECC_ERR_CTRL2

Address Offset 0x0000 1C1C

Description ECC Error Control2 Register
ECC Error Control2 Register.

Type RW

31	30	29	28	27	26	25	24	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																ECC_BIT2															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:0	ECC_BIT2	Data bit that needs to be flipped if double bit error needs to be forced	RW	0x0

Table 12-81. MCANSS_ECC_ERR_STAT1

Address Offset 0x0000 1C20

Description ECC Error Status1 Register
ECC Error Status1 Register.

Type RW

31	30	29	28	27	26	25	24	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																RESERVED						CLR_ECC_DED	CLR_ECC_SEC	RESERVED						ECC_DED	ECC_SEC

Bits	Field Name	Description	Type	Reset
31:16	ECC_ROW	Row address where the single or double-bit error has occurred	R	0x0

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x0
9	CLR_ECC_DED	Clear Double Bit Error Status	RW1TC	0x0
8	CLR_ECC_SEC	Clear Single Bit Error Status	RW1TC	0x0
7:2	RESERVED	Reserved	R	0x0
1	ECC_DED	Level Double Bit Error Status	RW1TS	0x0
0	ECC_SEC	Level Single Bit Error Status	RW1TS	0x0

Table 12-82. MCANSS_ECC_ERR_STAT2

Address Offset 0x0000 1C24

Description ECC Error Status2 Register
ECC Error Status2 Register.

Type R

31	30	29	28	27	26	25	24	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															

Bits	Field Name	Description	Type	Reset
31:16	ECC_BIT2	Data bit that corresponds to the double-bit error	R	0x0
15:0	ECC_BIT1	Data bit that corresponds to the single-bit error	R	0x0

Table 12-83. MCANSS_ECC_SEC_EOI_REG

Address Offset 0x0000 1C3C

Description EOI Register
EOI Register.

Type RW

31	30	29	28	27	26	25	24	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																															EOI_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EOI_WR	EOI Register	RW	0x0

Table 12-84. MCANSS_ECC_SEC_STATUS_REG0

Address Offset 0x0000 1C40

Description Interrupt Status Register 0
Interrupt Status Register 0.

Type RW

31	30	29	28	27	26	25	24	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																															MSGMEM_PEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	MSGMEM_PEND	Interrupt Pending Status for MSGMEM_PEND	RW1TS	0x0

Table 12-85. MCANSS_ECC_SEC_ENABLE_SET_REG0

Address Offset	0x0000 1C80
Description	Interrupt Enable Set Register 0 Interrupt Enable Set Register 0.
Type	RW

31	30	29	28	27	26	25	24	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																															MSGMEM_ENABLE_SET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_SET	Interrupt Enable Set Register for MSGMEM_PEND	RW1TS	0x0

Table 12-86. MCANSS_ECC_SEC_ENABLE_CLR_REG0

Address Offset	0x0000 1CC0
Description	Interrupt Enable Clear Register 0 Interrupt Enable Clear Register 0.
Type	RW

31	30	29	28	27	26	25	24	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																															MSGMEM_ENABLE_CLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_CLR	Interrupt Enable Clear Register for MSGMEM_PEND	RW1TC	0x0

Table 12-87. MCANSS_ECC_DED_EOI_REG

Address Offset	0x0000 1D3C
Description	EOI Register EOI Register.
Type	RW

31	30	29	28	27	26	25	24	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	EOI_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EOI_WR	EOI Register	RW	0x0

Table 12-88. MCANSS_ECC_DED_STATUS_REG0

Address Offset	0x0000 1D40
Description	Interrupt Status Register 0 Interrupt Status Register 0.
Type	RW

31	30	29	28	27	26	25	24	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																															MSGMEM_PEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_PEND	Interrupt Pending Status for MSGMEM_PEND	RW1TS	0x0

Table 12-89. MCANSS_ECC_DED_ENABLE_SET_REG0

Address Offset	0x0000 1D80
Description	Interrupt Enable Set Register 0 Interrupt Enable Set Register 0.
Type	RW

31	30	29	28	27	26	25	24	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																															MSGMEM_ENABLE_SET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_SET	Interrupt Enable Set Register for MSGMEM_PEND	RW1TS	0x0

Table 12-90. MCANSS_ECC_DED_ENABLE_CLR_REG0

Address Offset	0x0000 1DC0
Description	Interrupt Enable Clear Register 0 Interrupt Enable Clear Register 0.
Type	RW

31	30	29	28	27	26	25	24	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																															MSGMEM_ENABLE_CLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_CLR	Interrupt Enable Clear Register for MSGMEM_PEND	RW1TC	0x0

13 Multichannel Serial Port Interface (McSPI)

This chapter describes the McSPI of the device.

13.1 Introduction

This document is intended to provide programmers with a functional presentation of the Controller/Peripheral Multichannel Serial Port Interface (McSPI) module. It also provides a register description and a module configuration example.

McSPI is a general-purpose receive/transmit controller/peripheral controller that can interface with up to four peripheral external devices or one single external controller. It allows a duplex, synchronous, serial communication between a CPU and SPI compliant external devices (Peripherals and Controllers).

13.1.1 McSPI Features

The general features of the SPI controller are:

- Buffered receive/transmit data register per channel (1 word deep)
- Multiple SPI word access with one channel using a FIFO
- Two DMA requests per channel, one interrupt line
- Single interrupt line, for multiple interrupt source events
- Serial link interface supports:
  - Full duplex / Half duplex
  - Multi-channel controller or single channel peripheral operations
  - Programmable 1-32 bit transmit/receive shift operations.
  - Wide selection of SPI word lengths continuous from 4 to 32 bits
- Up to four SPI channels
- SPI word Transmit / Receive slot assignment based on round robin arbitration
- SPI configuration per channel (clock definition, enable polarity and word width)
- Clock generation supports:
  - Programmable controller clock generation (operating from fixed 48-MHz functional clock input)
  - Selectable clock phase and clock polarity per chip select.

13.1.2 Unsupported McSPI Features

This device supports only two chip selects per module. Module wakeup during peripheral mode operation is not supported, as noted in *McSPI Clock and Reset Management*.

Table 13-1. Unsupported McSPI Features

Feature	Reason
Chip selects 2 and 3	Not pinned out
Peripheral mode wakeup	SWAKEUP not connected
Retention during power down	Module not synthesized with retention enabled

13.2 Integration

This device includes two instantiations of McSPI: SPI0 and SPI1. The McSPI module is a general-purpose receive/transmit controller/peripheral controller that can interface with either up to four peripheral external devices or one single external controller. Figure 13-1 shows the example of a system with multiple external peripheral SPI compatible devices and Figure 13-2 shows the example of a system with an external controller.

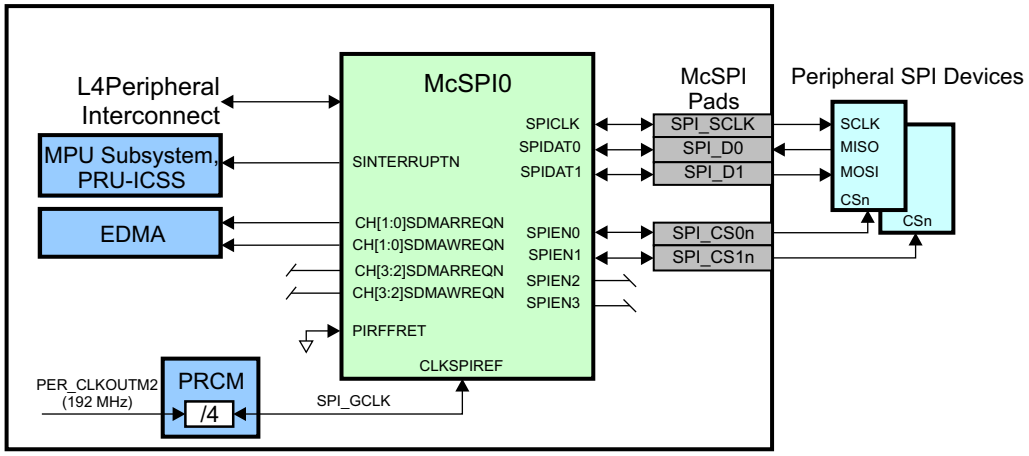


Figure 13-1. SPI Controller Application

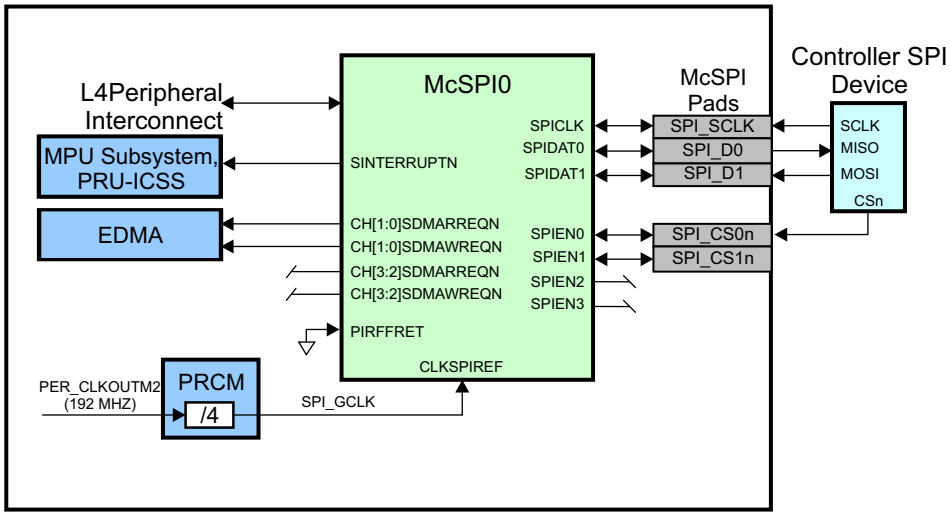


Figure 13-2. SPI Peripheral Application

ADVANCE INFORMATION

13.2.1 McSPI Connectivity Attributes

The general connectivity attributes for the McSPI module are shown in Table 13-2.

Table 13-2. McSPI Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (Interface/OCP) PD_PER_SPI_GCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU subsystem and PRU-ICSS (McSPI0INT) 1 interrupt to MPU subsystem only (McSPI1INT)
DMA Requests	4 DMA requests per instance to EDMA • 1 RX request for CS0 (SPIREVT0) • 1 TX request for CS0 (SPIXEVT0) • 1 RX request for CS1 (SPIREVT1) • 1 TX request for CS1 (SPIXEVT1)
Physical Address	L4 Peripheral port

13.2.2 McSPI Clock and Reset Management

The SPI module clocks can be woken up in two manners: by the SPI module itself using the SWAKEUP signal (refer to the module functional spec for detailed conditions), or directly from an external SPI controller device by detecting an active low level on its chip select input pin (CS0n) using a GPIO attached to that device pin. Neither of these methods is supported on the device.

Table 13-3. McSPI Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
CLKSPIREF Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_spi_gclk From PRCM

13.2.3 McSPI Pin List

The McSPI interface pins are summarized in Table 13-4.

Table 13-4. McSPI Pin List

Pin	Type	Description
SPIx_SCLK	I/O(1)	SPI serial clock (output when controller, input when peripheral)
SPIx_D0	I/O	Can be configured as either input or output (MOSI or MISO)
SPIx_D1	I/O	Can be configured as either input or output (MOSI or MISO)
SPIx_CS0	I/O	SPI chip select 0 output when controller, input when peripheral (active low)
SPIx_CS1	I/O	SPI chip select 1 output when controller, input when peripheral (active low)

(1) These signals are also used as inputs to re-time or sync data. The associated CONF_<module>_<pin>_RXACTIVE bit for these signals must be set to 1 to enable the inputs back to the module. It is also recommended to place a 33-ohm resistor in series (close to the processor) on each of these signals to avoid signal reflections.

13.3 Functional Description

13.3.1 SPI Transmission

This section describes the transmissions supported by McSPI. The SPI protocol is a synchronous protocol that allows a controller device to initiate serial communication with a peripheral device. Data is exchanged between these devices. A peripheral select line (SPIEN) can be used to allow selection of an individual peripheral SPI device. Peripheral devices that are not selected do not interfere with SPI bus activities. Connected to multiple external devices, McSPI exchanges data with a single SPI device at a time through two main modes:

- Two data pins interface mode. (See Section 13.3.1.1)
- Single data pin interface mode (recommended for half-duplex transmission). (See Section 13.3.1.2)

The flexibility of McSPI allows exchanging data with several formats through programmable parameters described in Section 13.3.1.3.

13.3.1.1 Two Data Pins Interface Mode

The two data pins interface mode, allows a full duplex SPI transmission where data is transmitted (shifted out serially) and received (shifted in serially) simultaneously on separate data lines SPIDAT [0] and SPIDAT [1]. Data leaving the controller exits on transmit serial data line also known as PICO: Peripheral IN Controller OUT. Data leaving the peripheral exits on the receive data line also known as POCI: Peripheral OUT Controller IN.

McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external controller/peripheral connection.

The serial clock (SPICLK) synchronizes shifting and sampling of the information on the two serial data lines (SPIDAT [1:0]). Each time a bit is transferred out from the Controller, one bit is transferred in from Peripheral.

Figure 13-3 shows an example of a full duplex system with a Controller device on the left and a Peripheral device on the right. After 8 cycles of the serial clock SPICLK, the WordA has been transferred from the controller to the peripheral. At the same time, the 8-bit WordB has been transferred from the peripheral to the controller.

When referring to the controller device, the control block transmits the clock SPICLK and the enable signal SPIEN (optional, see Section 13.7, *McSPI_MODULCTRL*).

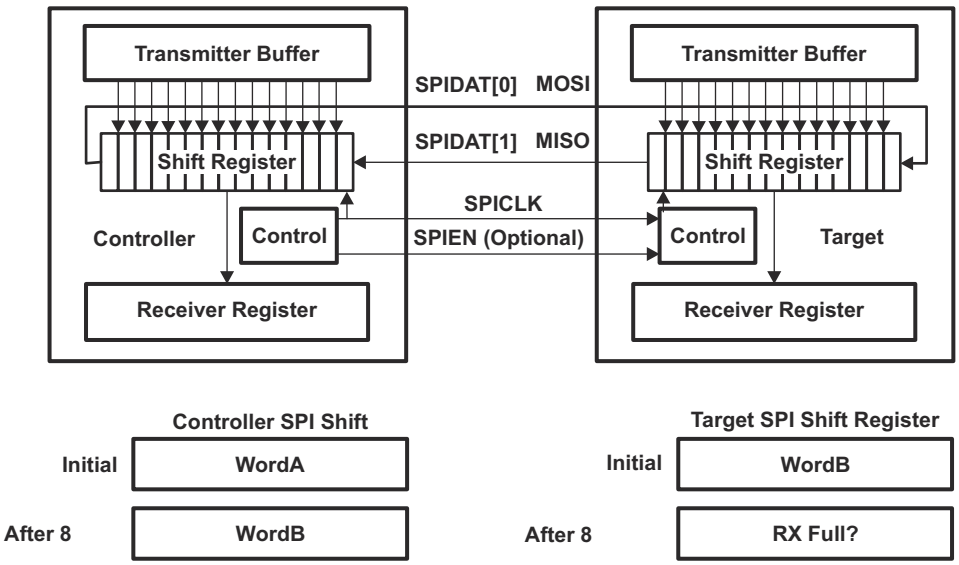


Figure 13-3. SPI Full-Duplex Transmission

13.3.1.2 Single Data Pin Interface Mode

In single data pin interface mode, under software control, a single data line is used to alternatively transmit and receive data (Half duplex transmission).

McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external controller/peripheral connection.

As for a full duplex transmission, the serial clock (SPICLK) synchronizes shifting and sampling of the information on the single serial data line.

13.3.1.2.1 Example With a Receive-Only Peripheral

Figure 13-4 shows a half duplex system with a Controller device on the left and a receive-only Peripheral device on the right. Each time a bit is transferred out from the Controller, one bit is transferred in the Peripheral. After 8 cycles of the serial clock SPICLK, the 8-bit WordA has been transferred from the controller to the peripheral.

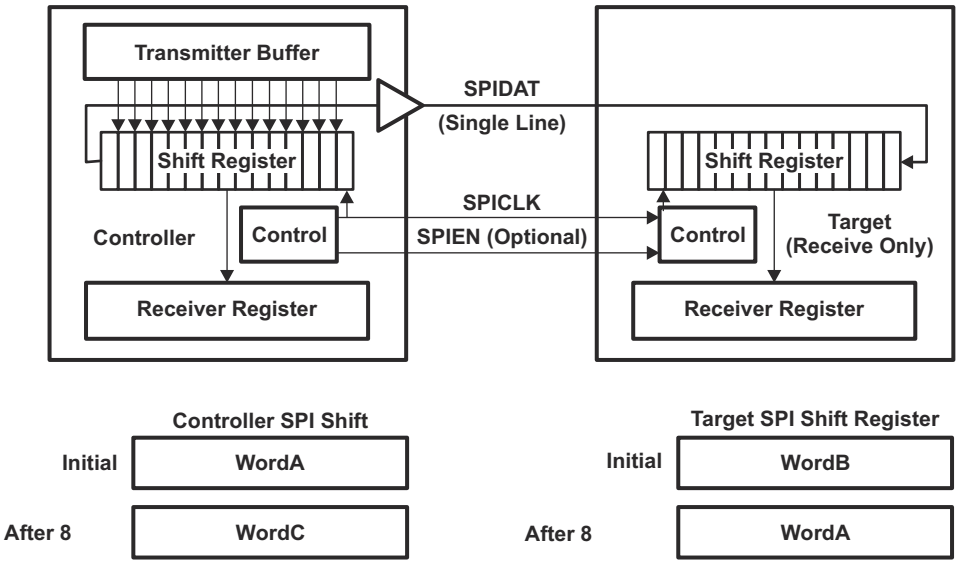


Figure 13-4. SPI Half-Duplex Transmission (Receive-only Peripheral)

13.3.1.2.2 Example With a Transmit-Only Peripheral

Figure 13-5 shows a half duplex system with a Controller device on the left and a transmit-only Peripheral device on the right. Each time a bit is transferred out from the Peripheral, one bit is transferred in the Controller. After 8 cycles of the serial clock SPICLK, the 8-bit WordA has been transferred from the peripheral to the controller.

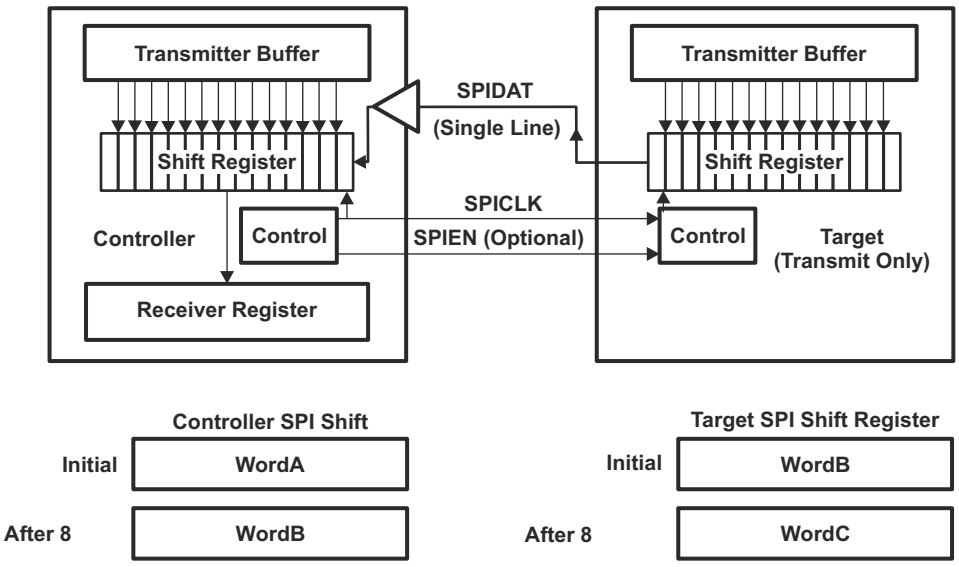


Figure 13-5. SPI Half-Duplex Transmission (Transmit-Only Peripheral)

ADVANCE INFORMATION

13.3.1.3 Transfer Formats

This section describes the transfer formats supported by McSPI.

The flexibility of McSPI allows setting the parameters of the SPI transfer:

- SPI word length
- SPI enable generation programmable
- SPI enable assertion
- SPI enable polarity
- SPI clock frequency
- SPI clock phase
- SPI clock polarity

The consistency between SPI word length, clock phase and clock polarity of the controller SPI device and the communicating peripheral device remains under software responsibility.

13.3.1.3.1 Programmable Word Length

McSPI supports any SPI word from 4 to 32 bits long.

The SPI word length can be changed between transmissions to allow a controller device to communicate with peripheral devices having different requirements.

13.3.1.3.2 Programmable SPI Enable Generation

McSPI is able to generate or not generate SPI enable. If management of chip select is de-asserted, a point-to-point connection is mandatory. Only a single controller of a peripheral device can be connected to the SPI bus.

13.3.1.3.3 Programmable SPI Enable (SPIEN)

The polarity of the SPIEN signals is programmable. SPIEN signals can be active high or low.

The assertion of the SPIEN signals is programmable. SPIEN signals can be manually asserted or automatically asserted.

Two consecutive words for two different peripheral devices may go along with active SPIEN signals with different polarity.

13.3.1.3.4 Programmable SPI Clock (SPICLK)

The phase and the polarity of the SPI serial clock are programmable when McSPI is a SPI controller device or a SPI peripheral device. The baud rate of the SPI serial clock is programmable when McSPI is a SPI controller.

When McSPI is operating as a peripheral, the serial clock SPICLK is an input from the controller.

13.3.1.3.5 Bit Rate

In Controller Mode, an internal reference clock CLKSPIREF is used as an input of a programmable divider to generate bit rate of the serial clock SPICLK. Granularity of this clock divider can be changed.

13.3.1.3.6 Polarity and Phase

McSPI supports four sub-modes of the SPI format transfer that depend on the polarity (POL) and the phase (PHA) of the SPI serial clock (SPICLK). Table 13-5 and Figure 13-6 show a summary of the four sub-modes. Software selects one of four combinations of serial clock phase and polarity.

Two consecutive SPI words for two different peripheral devices may go along with active SPICLK signal with different phase and polarity.

Table 13-5. Phase and Polarity Combinations

Polarity (POL)	Phase (PHA)	SPI Mode	Comments
0	0	mode0	SPICLK active high and sampling occurs on the rising edge.
0	1	mode1	SPICLK active high and sampling occurs on the falling edge.
1	0	mode2	SPICLK active low and sampling occurs on the falling edge.
1	1	mode3	SPICLK active low and sampling occurs on the rising edge.

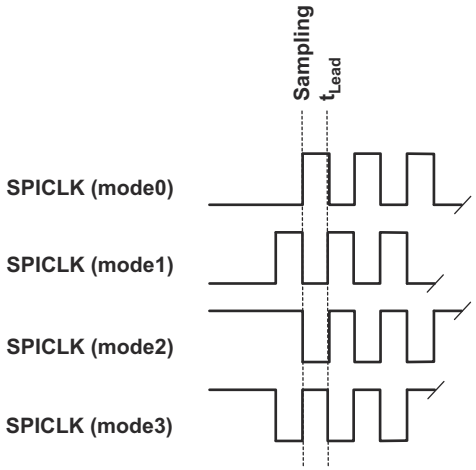


Figure 13-6. Phase and Polarity Combinations

13.3.1.3.7 Transfer Format With PHA = 0

This section describes the concept of a SPI transmission with the SPI mode0 and the SPI mode2.

In the transfer format with PHA = 0, SPIEN is activated a half cycle of SPICLK ahead of the first SPICLK edge.

In both controller and peripheral modes, McSPI drives the data lines at the time of SPIEN is asserted.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines, the first bit of SPI word is valid a half-cycle of SPICLK after the SPIEN assertion.

Therefore, the first edge of the SPICLK line is used by the controller to sample the first data bit sent by the peripheral. On the same edge, the first data bit sent by the controller is sampled by the peripheral.

On the next SPICLK edge, the received data bit is shifted into the shift register, and a new data bit is transmitted on the serial data line.

This process continues for a total of pulses on the SPICLK line defined by the SPI word length programmed in the controller device, with data being latched on odd numbered edges and shifted on even numbered edges.

Figure 13-7 is a timing diagram of a SPI transfer for the SPI mode0 and the SPI mode2, when McSPI is controller or peripheral, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in the data manual.

When McSPI is in peripheral mode, if the SPIEN line is not de-asserted between successive transmissions then the content of the Transmitter register is not transmitted, instead the last received SPI word is transmitted.

In controller mode, the SPIEN line must be negated and reasserted between each successive SPI word. This is because the peripheral select pin freezes the data in its shift register and does not allow it to be altered if PHA bit equals 0.

In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In peripheral mode, SPIEN is useless.

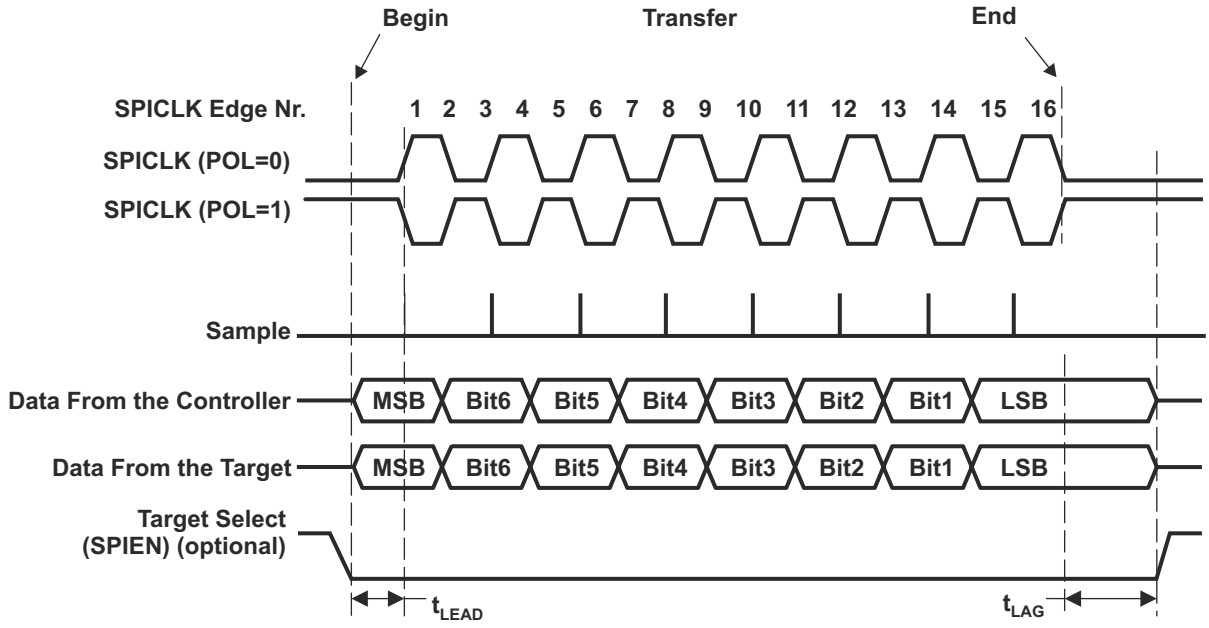


Figure 13-7. Full Duplex Single Transfer Format with PHA = 0

13.3.1.3.8 Transfer Format With PHA = 1

This section describes SPI full duplex transmission with the SPI mode1 and the SPI mode3.

In the transfer format with PHA = 1, SPIEN is activated a delay (t_{Lead}) ahead of the first SPICLK edge.

In both controller and peripheral modes, McSPI drives the data lines on the first SPICLK edge.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines, the first bit of SPI word is valid on the next SPICLK edge, a half-cycle later of SPICLK. It is the sampling edge for both the controller and peripheral.

When the third edge occurs, the received data bit is shifted into the shift register. The next data bit of the controller is provided to the serial input pin of the peripheral.

This process continues for a total of pulses on the SPICLK line defined by the word length programmed in the controller device, with data being latched on even numbered edges and shifted on odd numbered edges.

Figure 13-8 is a timing diagram of a SPI transfer for the SPI mode1 and the SPI mode3, when McSPI is controller or peripheral, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in the data manual.

The SPIEN line may remain active between successive transfers. In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In peripheral mode SPIEN is useless.

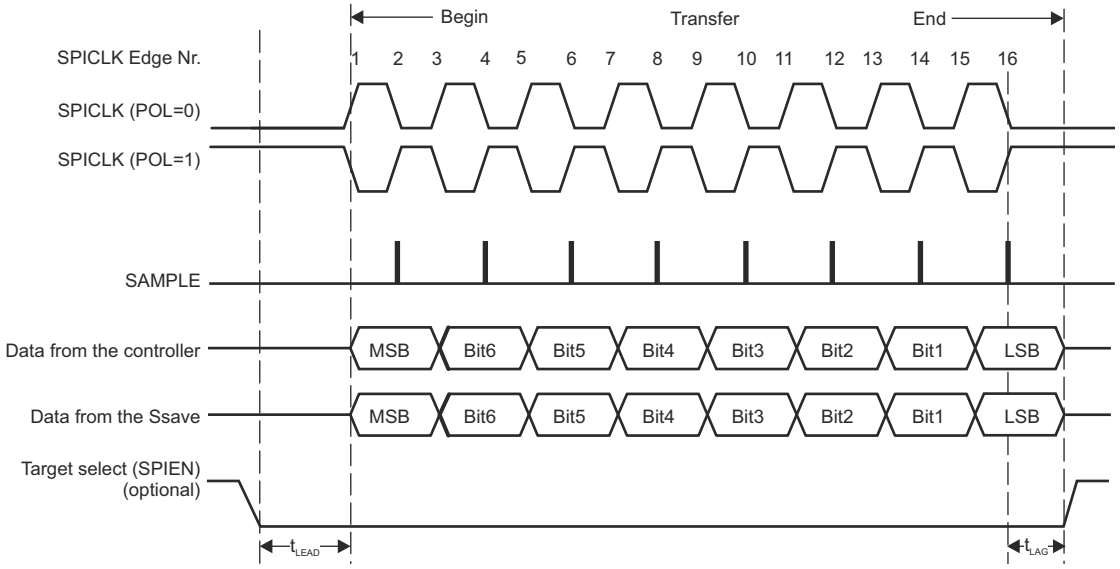


Figure 13-8. Full Duplex Single Transfer Format With PHA = 1

ADVANCE INFORMATION

13.3.2 Controller Mode

McSPI is in controller mode when the bit MS of the register MCSPI_MODULCTRL is cleared.

In controller mode McSPI supports multi-channel communication with up to 4 independent SPI communication channel contexts. McSPI initiates a data transfer on the data lines (SPIDAT [1:0]) and generates clock (SPICLK) and control signals (SPIEN) to a single SPI peripheral device at a time.

13.3.2.1 Dedicated Resources Per Channel

In the following sections, the letter “i” indicates the channel number that can be 0, 1, 2 or 3. Each channel has the following dedicated resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH(i)CTRL. Disabling the channel, outside data word transmission, remains under user responsibility.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH(i)STAT is set.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH(i)STAT is set.
- A fixed SPI ENABLE line allocation (SPIEN[i] port for channel "i"), SPI enable management is optional.
- Its own communication configuration with the following parameters via the register MCSPI_CH(i)CONF
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - SPIEN polarity, programmable with the bit EPOL.
  - SPIEN kept active between words, programmable with the bit FORCE.
  - Turbo mode, programmable with the bit TURBO.
  - SPICLK frequency, programmable with the bit CLKD, the granularity of clock division can be changed using CLKG bit, the clock ratio is then concatenated with MCSPI_CH(i)CTRL[EXTCLK] value.
  - SPICLK polarity, programmable with the bit POL
  - SPICLK phase, programmable with the bit PHA.
  - Start bit polarity, programmable with the bit SBPOL
  - Use a FIFO Buffer or not (see the following note), programmable with FFER and FFEW, depending on transfer mode, (MCSPI_CH(i)CONF[TRM]).
- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW.
- Three interrupts events

Note: When more than one channel has an FIFO enable bit field (FFER or FFEW) set, the FIFO will not be used on any channel. Software must ensure that only one enabled channel is configured to use the FIFO buffer.

The transfers will use the latest loaded parameters of the register MCSPI_CH(i)CONF.

The configuration parameters SPIEN polarity, Turbo mode, SPICLK phase and SPICLK polarity can be loaded in the MCSPI_CH(i)CONF register only when the channel is disabled. The user has the responsibility to change the other parameters of the MCSPI_CH(i)CONF register when no transfer occurs on the SPI interface.

13.3.2.2 Interrupt Events in Controller Mode

In controller mode, the interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt event related to the receiver register state is RX_full.

13.3.2.2.1 TX_empty

The event TX_empty is activated when a channel is enabled and its transmitter register becomes empty (transient event). Enabling channel automatically raises this event, except for the Controller receive only mode. (See Section 13.3.2.5). When the FIFO buffer is enabled (MCSPI_CH(i)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in the buffer to write a number of bytes defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be loaded to remove the source of the interrupt and the TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source) . (See Section 13.3.4).

When FIFO is enabled, no new TX_empty event will be asserted as soon as CPU has not performed the number of writes into the transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of CPU to perform the right number of writes.

13.3.2.2.2 TX_underflow

The event TX_underflow is activated when the channel is enabled and if the transmitter register or FIFO is empty (not updated with new data) at the time of shift register assignment.

The TX_underflow is a harmless warning in controller mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

To avoid having a TX_underflow event, the Transmit Register (MCSPI_TX(i)) should be loaded as infrequently as possible.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

Note: When more than one channel has an FIFO enable bit field (FFER or FFEW) set, the FIFO will not be used on any channel. Software must ensure that only one enabled channel is configured to use the FIFO buffer.

13.3.2.2.3 RX_ full

The event RX_full is activated when channel is enabled and receiver register becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(i)CONF[FFER] set to 1), the RX_full is asserted when the number of bytes in the buffer equals the level defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source).

When the FIFO is enabled, no new RX_FULL event will be asserted once the CPU has read the number of bytes defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of the CPU to perform the correct number of read operations.

13.3.2.2.4 End of Word Count

The event end of word (EOW) count is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfers defined in the MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The EOW count interrupt also indicates that the SPI transfer has halted on the channel using the FIFO buffer.

The EOW interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

13.3.2.3 Controller Transmit and Receive Mode

This mode is programmable per channel (bit TRM of register MCSPI_CH(i)CONF).

The channel access to the shift registers, for transmission/reception, is based on its transmitter and receiver register state and round robin arbitration.

The channel that meets the rules below is included in the round robin list of already active channels scheduled for transmission and/or reception. The arbiter skips the channel that does not meet the rules and search for the next following enabled channel, in rotation.

Rule 1: Only enabled channels (bit EN of the register MCSPI_CH(i)CTRL), can be scheduled for transmission and/or reception.

Rule 2: An enabled channel can be scheduled if its transmitter register is not empty (bit TXS of the register MCSPI_CH(i)STAT) or its FIFO is not empty when the buffer is used for the corresponding channel (bit FFE of the register MCSPI_CH(i)STAT) at the time of shift register assignment. If the transmitter register or FIFO is empty, at the time of shift register assignment, the event TX_underflow is activated and the next enabled channel with new data to transmit is scheduled. (See also transmit only mode).

Rule 3: An enabled channel can be scheduled if its receive register is not full (bit RXS of the register MCSPI_CH(i)STAT)) or its FIFO is not full when the buffer is used for the corresponding channel (bit FFF of the register MCSPI_CH(i)STAT) at the time of shift register assignment. (See also receive only mode). Therefore the receiver register of FIFO cannot be overwritten. The RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) the updated transmitter register for the next scheduled channel is loaded into the shift register. This bit is meaningless when using the Buffer for this channel. The serialization (transmit and receive) starts according to the channel communication configuration. On serialization completion the received data is transferred to the channel receive register.

The built-in FIFO is available in this mode and if configured in one data direction, transmit or receive, then the FIFO is seen as a unique 64-byte buffer. If configured in both data directions, transmit and receive, then the FIFO is split into two separate 32-byte buffers with their own address space management. In this last case, the definition of AEL and AFL levels is based on 32 bytes and is under CPU responsibility.

13.3.2.4 Controller Transmit-Only Mode

This mode eliminates the need for the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

The controller transmit only mode is programmable per channel (bits TRM of the register MCSPI_CH(i)CONF).

In controller transmit only mode, transmission starts after data is loaded into the transmitter register.

Rule 1 and **Rule 2**, defined above, are applicable in this mode.

Rule 3, defined above, is not applicable: In controller transmit only mode, the receiver register or FIFO state “full” does not prevent transmission, and the receiver register is always overwritten with the new SPI word. This event in the receiver register is not significant when only transmission is meaningful. So, the RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

The McSPI module automatically disables the RX_full interrupt status. The corresponding interrupt request and DMA Read request are not generated in controller transmit only mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(i)STAT. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFEW bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

13.3.2.5 Controller Receive-Only Mode

This mode eliminates the need for the CPU to refill the transmitter register (minimizing data movement) when only reception is meaningful.

The controller receive mode is programmable per channel (bits TRM of the register MCSPI_CH(i)CONF).

The controller receive only mode enables channel scheduling only on empty state of the receiver register.

Rule 1 and **Rule 3**, defined above, are applicable in this mode.

Rule 2, defined above, is not applicable: In controller receive only mode, after the first loading of the transmitter register of the enabled channel, the transmitter register state is maintained as full. The content of the transmitter register is always loaded into the shift register, at the time of shift register assignment. So, after the first loading of the transmitter register, the bits TX_empty and TX_underflow, in the MCSPI_IRQSTATUS register are never set in this mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(i)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

13.3.2.6 Single-Channel Controller Mode

When the SPI is configured as a controller device with a single enabled channel, the assertion of the SPIM_CSX signal can be controlled in two different ways:

- In 3 pin mode : MCSPI_MODULCTRL[1] PIN34 and MCSPI_MODULCTRL[0] SINGLE bit are set to 1, the controller transmit SPI word as soon as transmit register or FIFO is not empty.
- In 4 pin mode : MCSPI_MODULCTRL[1] PIN34 bit is cleared to 0 and MCSPI_MODULCTRL[0] SINGLE bit is set to 1, SPIEN assertion/deassertion controlled by Software. (See Section 13.3.2.6.1) using the MCSPI_CH(i)CONF[20] FORCE bit.

13.3.2.6.1 Programming Tips When Switching to Another Channel

When a single channel is enabled and data transfer is ongoing:

- Wait for completion of the SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) before disabling the current channel and enabling a different channel.
- Disable the current channel first, and then enable the other channel.

13.3.2.6.2 Keep SPIEN Active Mode (Force SPIEN)

Continuous transfers are manually allowed by keeping the SPIEN signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the SPIEN line. This mode is supported by all channels and any controller sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the SPIEN active mode is supported when:

- A single channel is used (bit MCSPI_MODULCTRL[Single] is set to 1).
- Transfer parameters of the transfer are loaded in the configuration register (MCSPI_CH(i)CONF) in the appropriate channel.

  The state of the SPIEN signal is programmable.
  - Writing 1 into the bit FORCE of the register MCSPI_CH(i)CONF drives high the SPIEN line when MCSPI_CH(i)CONF[EPOL] is set to zero, and drives it low when MCSPI_CH(i)CONF[EPOL] is set.
  - Writing 0 into the bit FORCE of the register MCSPI_CH(i)CONF drives low the SPIEN line when MCSPI_CH(i)CONF[EPOL] is set to zero, and drives it high when MCSPI_CH(i)CONF[EPOL] is set.
- A single channel is enabled (MCSPI_CH(i)CTRL[En] set to 1) . The first enabled channel activates the SPIEN line.

Once the channel is enabled, the SPIEN signal is activated with the programmed polarity.

As in multi-channel controller mode, the start of the transfer depends on the status of the transmitter register, the status of the receiver register and the mode defined by the bits TRM in the configuration register (transmit only, receive only or transmit and receive) of the enabled channel.

The status of the serialization completion of each SPI word is given by the bit EOT of the register MCSPI_CH(i)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register.

A change in the configuration parameters is propagated directly on the SPI interface. If the SPIEN signal is activated the user must insure that the configuration is changed only between SPI words, in order to avoid corrupting the current transfer.

Note

The SPIEN polarity, the SPICLK phase and SPICLK polarity must not be modified when the SPIEN signal is activated. The Transmit/Receive mode, programmable with the bit TRM can be modified only when the channel is disabled. The channel can be disabled and enabled while the SPIEN signal is activated.

The delay between SPI words that requires the connected SPI peripheral device to switch from one configuration (transmit only for instance) to another (receive only for instance) must be handled under software responsibility.

At the end of the last SPI word, the channel must be deactivated (MCSPI_CH(i)CTRL[En] is cleared to 0) and the SPIEN can be forced to its inactive state (MCSPI_CH(i)CONF[Force]).

Figure 13-9 and Figure 13-10 show successive transfers with SPIEN kept active low with a different configuration for each SPI word in respectively single data pin interface mode and two data pins interface mode. The arrows indicate when the channel is disabled before a change in the configuration parameters and enabled again.

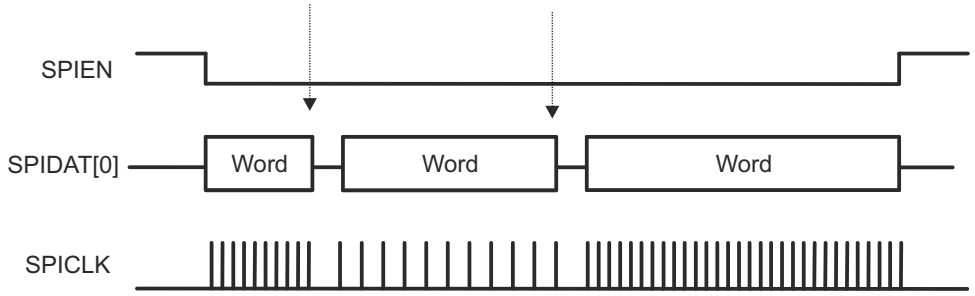


Figure 13-9. Continuous Transfers With SPIEN Maintained Active (Single-Data-Pin Interface Mode)

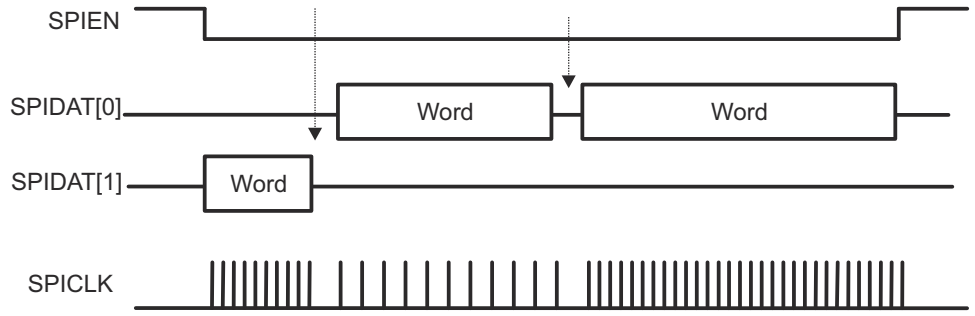


Figure 13-10. Continuous Transfers With SPIEN Maintained Active (Dual-Data-Pin Interface Mode)

Note

The turbo mode is also supported for the Keep SPIEN active mode when the following conditions are met:

- A single channel will be explicitly used (bit MCSPI_MODULCTRL[Single] is set to 1).
- The turbo mode is enabled in the configuration of the channel (bit Turbo of the register MCSPI_CH(i)CONF).

ADVANCE INFORMATION

13.3.2.6.3 Turbo Mode

The purpose of the Turbo mode is to improve the throughput of the SPI interface when a single channel is enabled, by allowing transfers until the shift register and the receiver register are full.

This mode is programmable per channel (bit Turbo of the register MCSPI_CH(i)CONF). When several channels are enabled, the bit Turbo of the registers MCSPI_CH(i)CONF has no effect, and the channel access to the shift registers remains as described in Section 13.3.2.3.

In Turbo mode, **Rule 1** and **Rule 2** defined in Section 13.3.2.3 are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(i)STAT) at the time of shift register assignment until the shift register is full.

In Turbo mode, **Rule 1** and **Rule 2** defined in Section 13.3.2.3 are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(i)STAT) at the time of shift register assignment until the shift register is full.

The receiver register cannot be overwritten in Turbo mode. In consequence the RX_overflow bit, in MCSPI_IRQSTATUS register is never set in this mode.

13.3.2.7 Start Bit Mode

The purpose of the start bit mode is to add an extended bit before the SPI word transmission specified by word length WL. This feature is only available in controller mode.

This mode is programmable per channel using the start bit enable (SBE) bit of the register MCSPI_CH(i)CONF).

The polarity of the extended bit is programmable per channel and it indicates whether the next SPI word must be handled as a command when SBPOL is cleared to 0 or as a data or a parameter when SBPOL is set to 1. Moreover start bit polarity SBPOL can be changed dynamically during start bit mode transfer without disabling the channel for reconfiguration, in this case you have the responsibility to configure the SBPOL bit before writing the SPI word to be transmitted in TX register.

The start bit mode could be used at the same time as turbo mode and/or manual chip select mode. In this case only one channel could be used, no round-robin arbitration is possible.

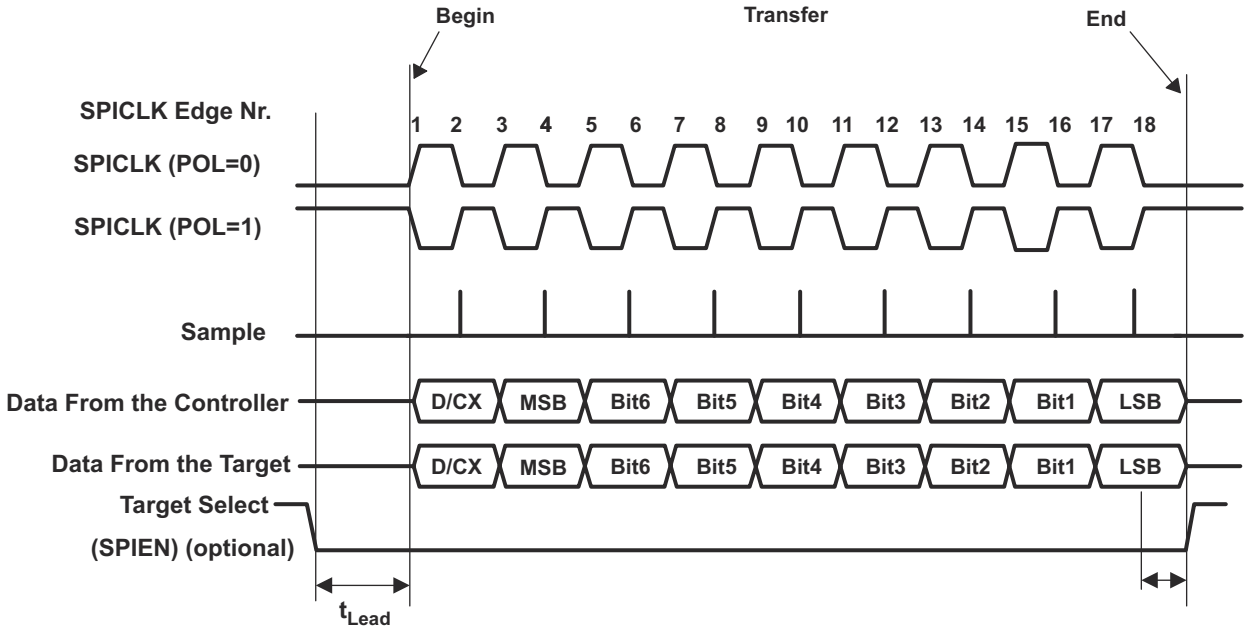


Figure 13-11. Extended SPI Transfer With Start Bit PHA = 1

13.3.2.8 Chip-Select Timing Control

The chip select timing control is only available in controller mode with automatic chip select generation (FORCE bit field is cleared to 0), to add a programmable delay between chip select assertion and first clock edge or chip select removal and last clock edge. The option is available only in 4 pin mode MCSPI_MODULCTRL[1] PIN34 is cleared to 0.

This mode is programmable per channel (bit TCS of the register MCSPI_CH(i)CONF). Figure 13-12 shows the chip-select SPIEN timing control.

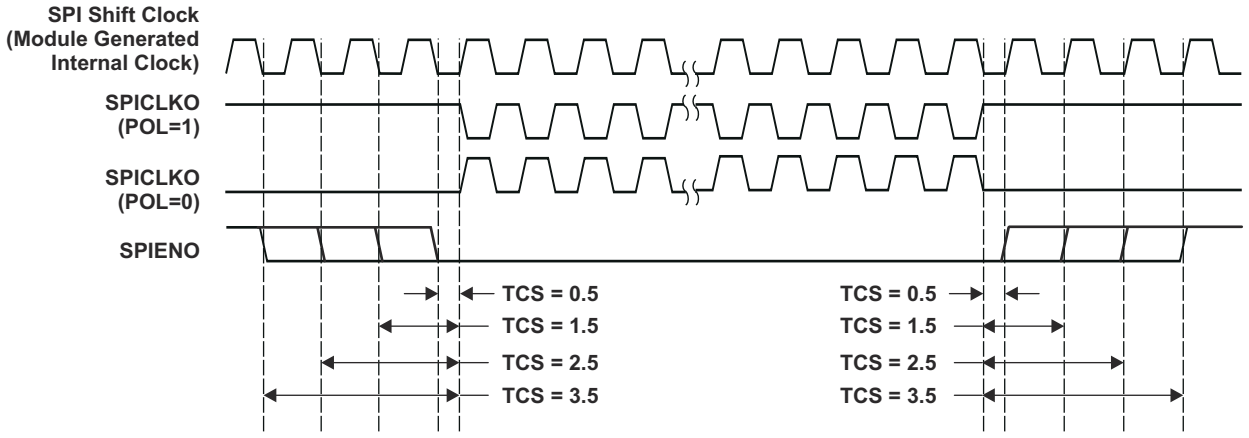


Figure 13-12. Chip-Select SPIEN Timing Controls

Note

Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between chip-select assertion and the first clock edge with PHA = 1 or between chip-select removal and the last clock edge with PHA = 0.

With an odd clock divider ratio which occurs when granularity is one clock cycle, that means that MCSPI_CH(i)CONF[CLKG] is set to 1 and MCSPI_CH(i)CONF[CLKD] has an even value, the clock duty cycle is not 50%, then one of the high level or low level duration is selected to be added to TCS delay.

Table 13-6 summarizes all delays between chip select and first (setup) or last (hold) clock edge.

In 3-pin mode this option is useless, the chip select SPIEN is forced to low state.

Table 13-6. Chip Select ↔ Clock Edge Delay Depending on Configuration

Clock Ratio F_{ratio}	Clock Phase PHA	Chip Select ↔ Clock Edge Delay	
		Setup	Hold
1	0	T_ref × (TCS + ½)	T_ref × (TCS + 1)
	1	T_ref × (TCS + 1)	T_ref × (TCS + ½)
Even ≥ 2	x	T_ref × F_{ratio} × (TCS + ½)	T_ref × F_{ratio} × (TCS + ½)
Odd ≥ 3 (only with MCSPI_CH(i)CONF[CLKG] set to 1	0	T_ref × [{F_{ratio} × TCS) + (F_{ratio} + ½)]	T_ref × [{F_{ratio} × TCS) + (F_{ratio} + ½)]
	1	T_ref × [{F_{ratio} × TCS) + (F_{ratio} - ½)]	T_ref × [{F_{ratio} × TCS) + (F_{ratio} - ½)]

T_ref = CLKSPIREF period in ns. F_{ratio} = SPI clock division ratio

The clock divider ratio depends on divider granularity MCSPI_CH(i)CONF[CLKG]:

- MCSPI_CH(i)CONF[CLKG] = 0 : granularity is power of two.

  $F_{ratio} = 2^{MCSPI\_CH(i)CONF[CLKD]}$
- MCSPI_CH(i)CONF[CLKG] = 1 : granularity is one cycle.

  F_{ratio} = MCSPI_CH(i)CTRL[EXTCLK] × MCSPI_CH(i)CONF[CLKD] + 1

13.3.2.9 Clock Ratio Granularity

By default the clock division ratio is defined by the register MCSPI_CH(i)CONF[CLKD] with power of two granularity leading to a clock division in range 1 to 32768, in this case the duty cycle is always 50%. With bit MCSPI_CH(i)CONF[CLKG] the clock division granularity can be changed to one clock cycle, in that case the register MCSPI_CH(i)CTRL[EXTCLK] is concatenated with MCSPI_CH(i)CONF[CLKD] to give a 12-bit width division ratio in range 1 to 4096.

When granularity is one clock cycle (MCSPI_CH(i)CONF[CLKG] set to 1), for odd value of clock ratio the clock duty cycle is kept to 50-50 using falling edge of clock reference CLKSPIREF.

Table 13-7. CLKSPIO High/Low Time Computation

Clock Ratio F_{ratio}	CLKSPIO High Time	CLKSPIO Low Time
1	T_{high}_ref	T_{low}_ref
Even ≥ 2	t_ref × (F_{ratio}/2)	t_ref × (F_{ratio}/2)
Odd ≥ 3	t_ref × (F_{ratio}/2)	t_ref × (F_{ratio}/2)

T_ref = CLKSPIREF period in ns. T_{high}_ref = CLKSPIREF high Time period in ns. T_{low}_ref = CLKSPIREF low Time period in ns. F_{ratio} = SPI clock division ratio

F_{ratio} = MCSPI_CH(i)CTRL[EXTCLK] × MCSPI_CH(i)CONF[CLKD] + 1

For odd ratio value the duty cycle is calculated as below:

Duty_cycle = ½

Granularity examples: With a clock source frequency of 48 MHz:

Table 13-8. Clock Granularity Examples

MCSPI_CH(i)CTRL	MCSPI_CH(i)CONF	MCSPI_CH(i)CONF		MCSPI_CH(i)CONF	MCSPI_CH(i)CONF	Thigh (ns)	Tlow (ns)	Tperiod (ns)	Duty Cycle	Fout (MHz)
EXTCLK	CLKD	CLKG	F_{ratio}	PHA	POL					
X	0	0	1	X	X	10.4	10.4	20.8	50-50	48
X	1	0	2	X	X	20.8	20.8	41.6	50-50	24
X	2	0	4	X	X	41.6	41.6	83.2	50-50	12
X	3	0	8	X	X	83.2	83.2	166.4	50-50	6
0	0	1	1	X	X	10.4	10.4	20.8	50-50	48
0	1	1	2	X	X	20.8	20.8	41.6	50-50	24
0	2	1	3	1	0	31,2	31,2	62.4	50-50	16
0	2	1	3	1	1	31,2	31,2	62.4	50-50	16
0	3	1	4	X	X	41.6	41.6	83.2	50-50	12
5	0	1	81	1	0	842,4	842,4	1684.8	50-50	0.592
5	7	1	88	X	X	915.2	915.2	1830.4	50-50	0.545

ADVANCE INFORMATION

13.3.2.10 FIFO Buffer Management

The McSPI controller has a built-in 64-byte buffer in order to unload DMA or interrupt handler and improve data throughput.

This buffer can be used by only one channel and is selected by setting MCSPI_CH(i)CONF[FFER] and/or MCSPI_CH(i)CONF[FFEW] to 1.

If several channels are selected and several FIFO enable bit fields set to 1, the controller forces the buffer to be disabled for all channels. It is the responsibility of the driver to enable the buffer for only one channel.

The buffer can be used in the modes defined below:

- Controller or Peripheral mode.
- Transmit only, Receive only or Transmit/Receive mode.
- Single channel or turbo mode, or in normal round robin mode. In round robin mode the buffer is used by only one channel.
- All word length MCSPI_CH(i)CONF[WL] are supported.

Two levels AEL and AFL located in MCSPI_XFERLEVEL register rule the buffer management. The granularity of these levels is one byte, then it is not aligned with SPI word length. It is the responsibility of the driver to set these values as a multiple of SPI word length defined in MCSPI_CH(i)CONF[WL]. The number of byte written in the FIFO depends on word length (see Table 13-9).

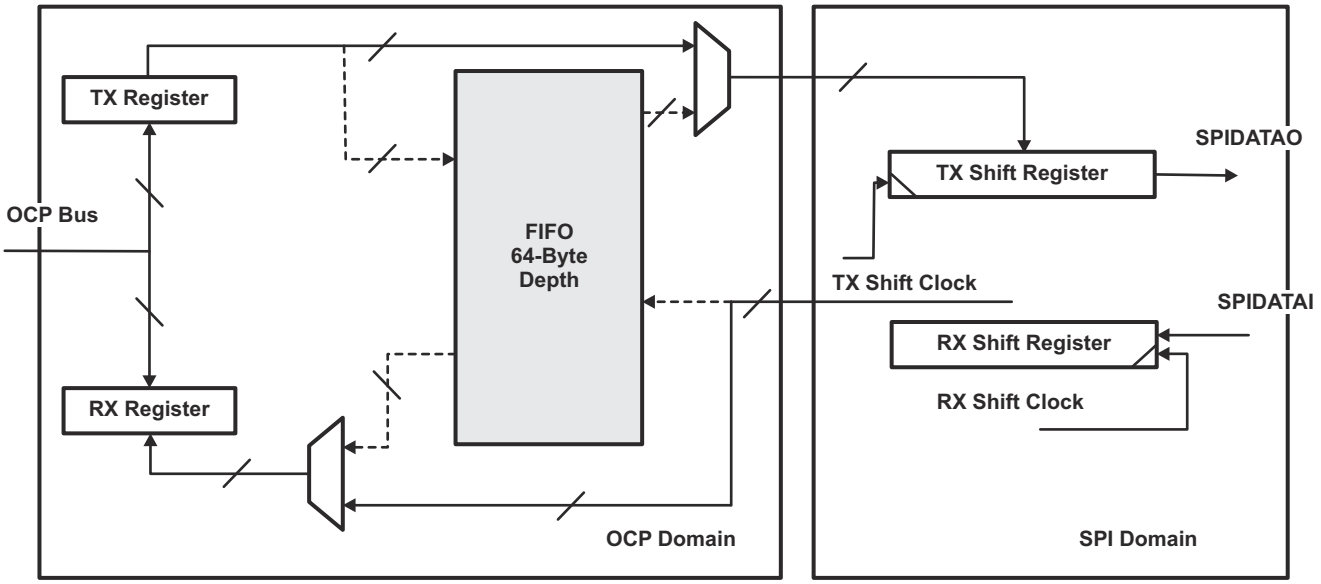
Table 13-9. FIFO Writes, Word Length Relationship

	SPI Word Length WL		
	3 ≤ WL ≤ 7	**8 ≤ WL ≤ 15**	**16 ≤ WL ≤ 31**
Number of byte written in the FIFO	1 byte	2 bytes	4 bytes

13.3.2.10.1 Split FIFO

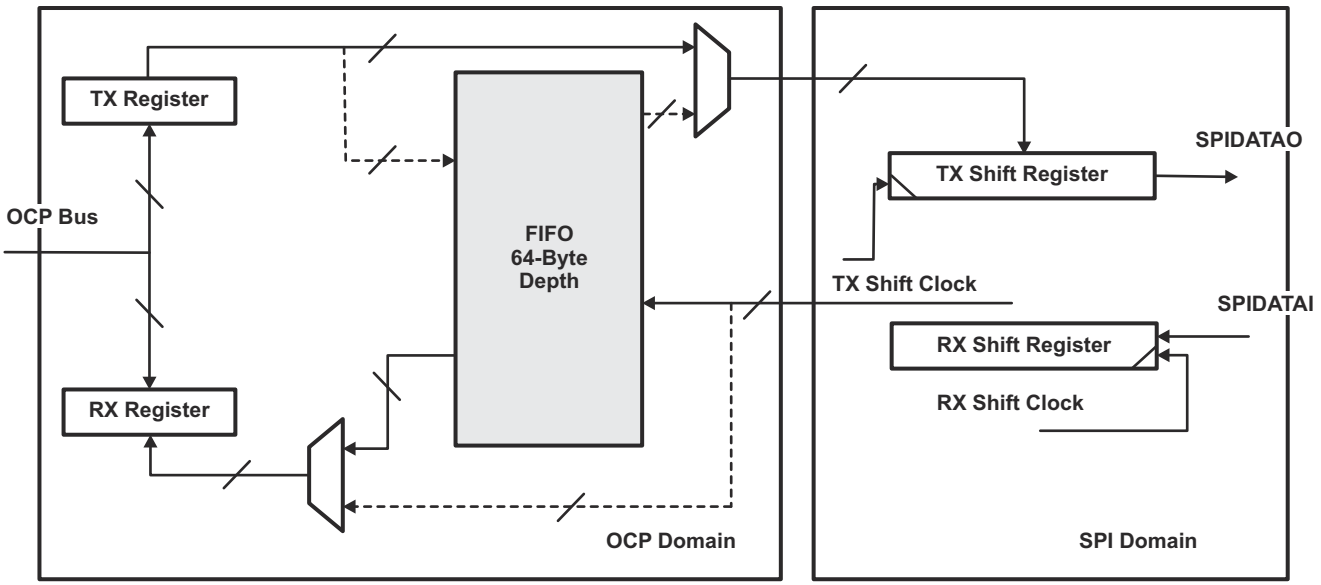
The FIFO can be split into two part when module is configured in transmit/receive mode MCSPI_CH(i)CONF[TRM] is cleared to 0 and MCSPI_CH(i)CONF[FFER] and MCSPI_CH(i)CONF[FFEW] asserted. Then system can access a 32-byte depth FIFO per direction.

The FIFO buffer pointers are reset when the corresponding channel is enabled or FIFO configuration changes.



Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

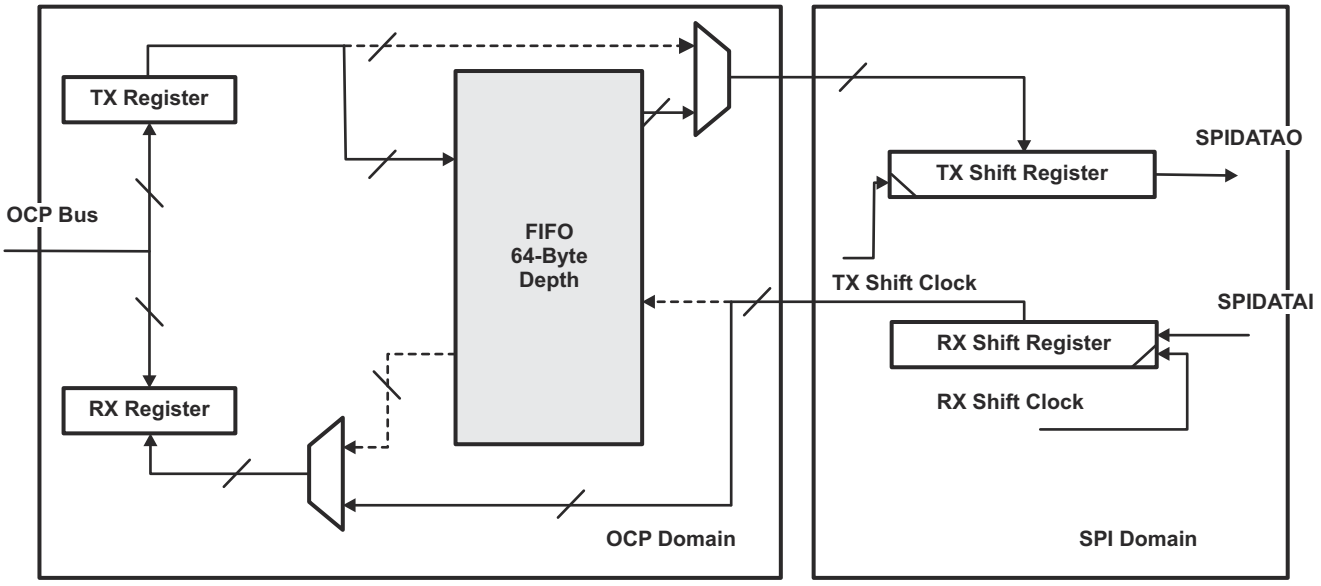
Figure 13-13. Transmit/Receive Mode With No FIFO Used



Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

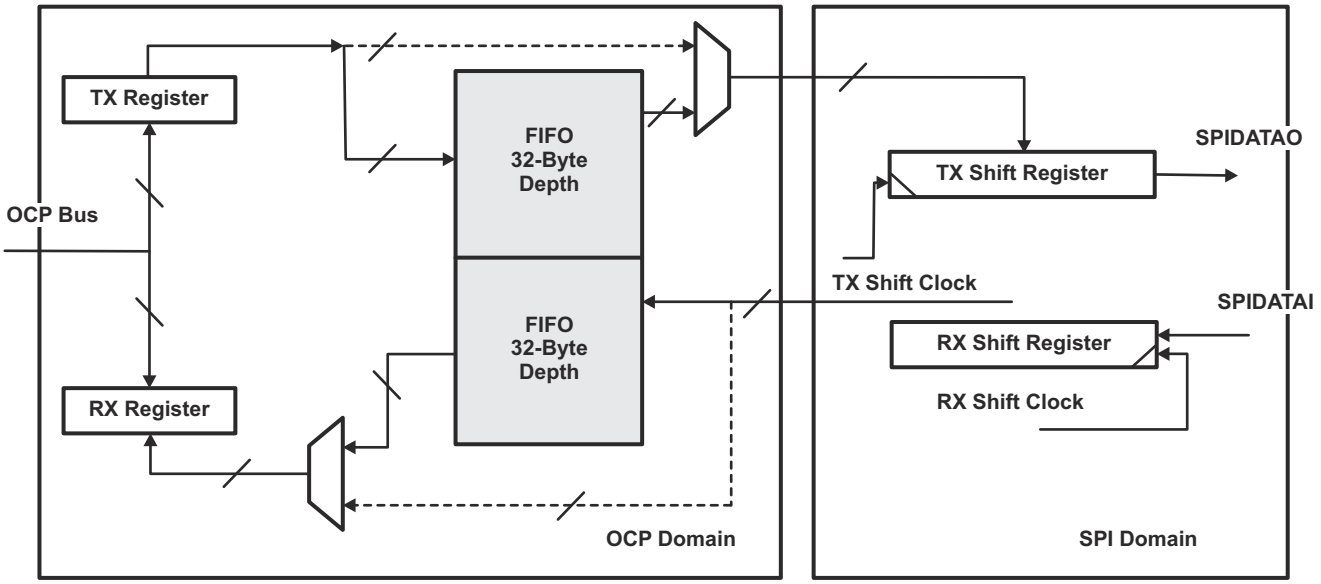
Figure 13-14. Transmit/Receive Mode With Only Receive FIFO Enabled

ADVANCE INFORMATION



Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x1 FIFO enabled on transmit path

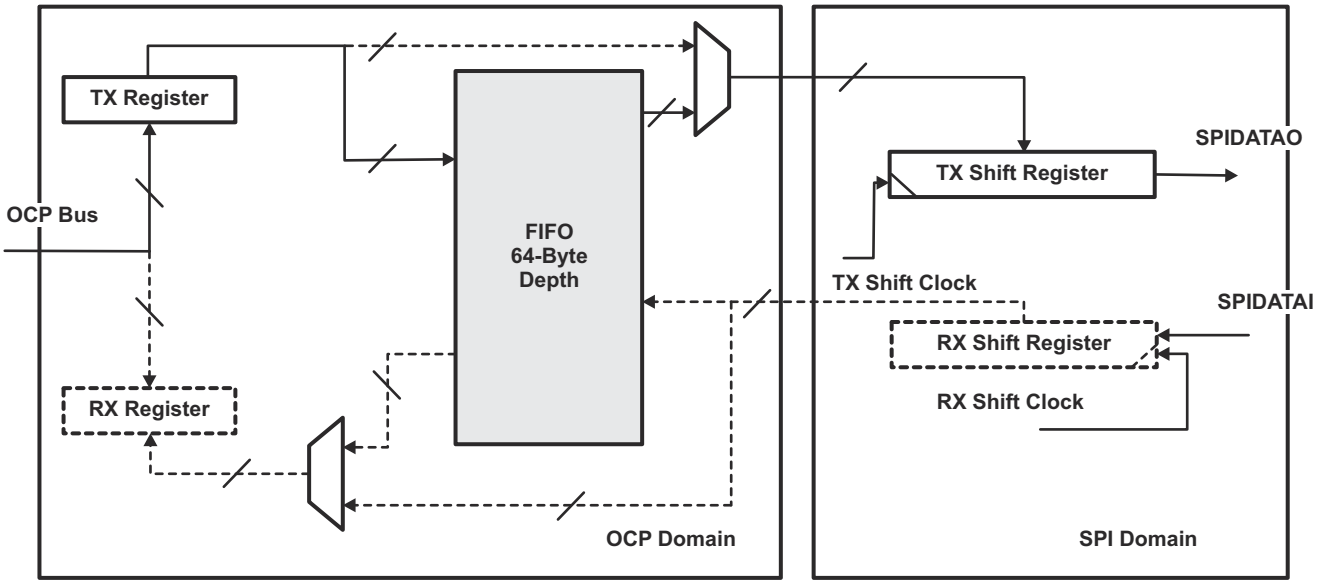
Figure 13-15. Transmit/Receive Mode With Only Transmit FIFO Used



Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

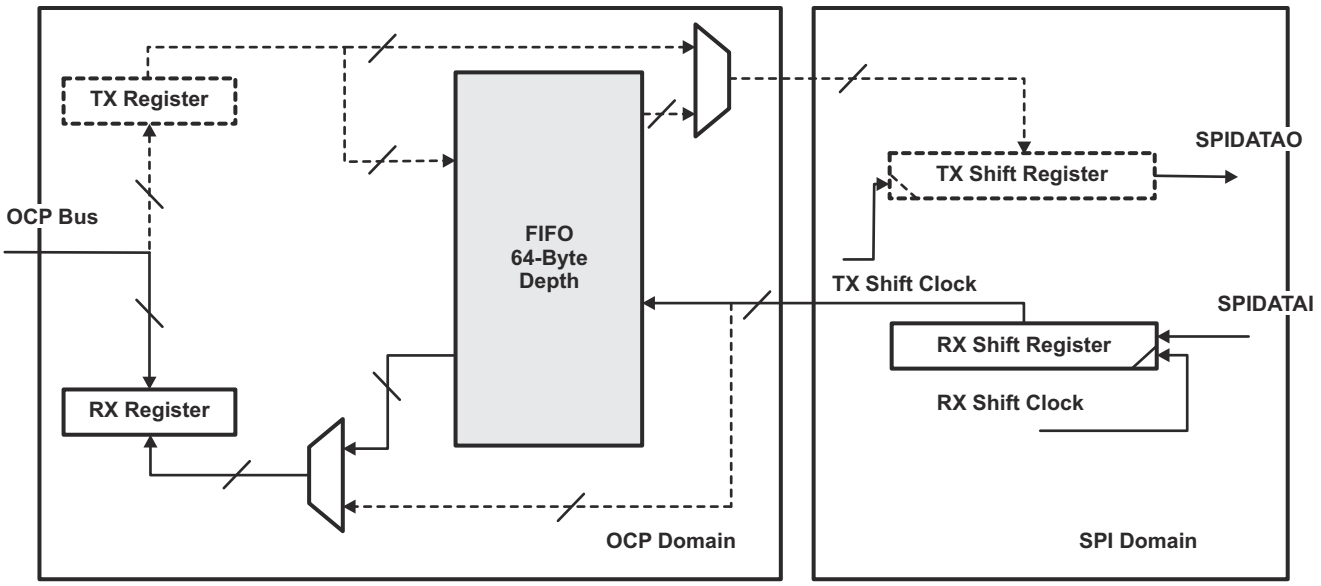
Figure 13-16. Transmit/Receive Mode With Both FIFO Direction Used

ADVANCE INFORMATION



Configuration:
MCSPI_CH(i)CONF[TRM]=0x2 Transmit only mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on transmit path
MCSPI_CH(i)CONF[FFWE] not applicable

Figure 13-17. Transmit-Only Mode With FIFO Used



Configuration:
MCSPI_CH(i)CONF[TRM]=012 Receive only mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE] not applicable

Figure 13-18. Receive-Only Mode With FIFO Used

13.3.2.10.2 Buffer Almost Full

The bit field MCSPI_XFERLEVEL[AFL] is needed when the buffer is used to receive SPI word from a peripheral (MCSPI_CH(i)CONF[FFER] must be set to 1). It defines the almost full buffer status.

When FIFO pointer reaches this level an interrupt or a DMA request is sent to the CPU to enable system to read AFL+1 bytes from receive register. Be careful AFL+1 must correspond to a multiple value of MCSPI_CH(i)CONF[WL].

When DMA is used, the request is de-asserted after the first receive register read.

No new request will be asserted until the system has performed the correct number of read operations from the buffer.

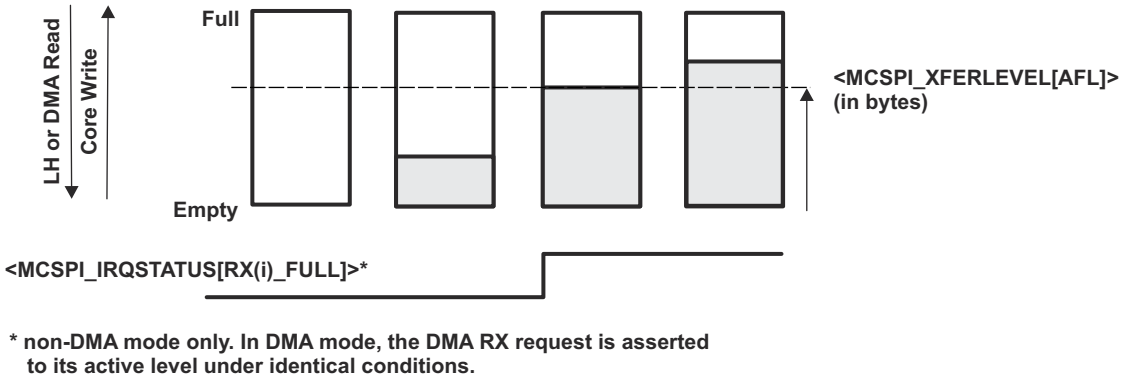


Figure 13-19. Buffer Almost Full Level (AFL)

Note

SPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_RX*n* request is asserted on the same conditions as the SPI_IRQSTATUS RX*n*_FULL flag.

ADVANCE INFORMATION

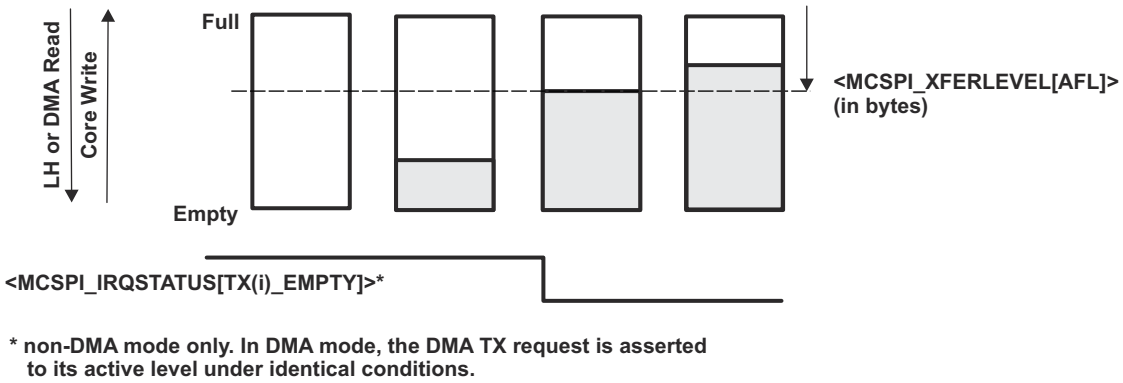
13.3.2.10.3 Buffer Almost Empty

The bitfield MCSPI_XFERLEVEL[AEL] is needed when the buffer is used to transmit SPI word to a peripheral (MCSPI_CH(i)CONF[FFEW]must be set to 1). It defines the almost empty buffer status.

When FIFO pointer has not reached this level an interrupt or a DMA request is sent to the CPU to enable system to write AEL+1 bytes to transmit register. Be careful AEL+1 must correspond to a multiple value of MCSPI_CH(i)CONF[WL].

When DMA is used, the request is de-asserted after the first transmit register write.

No new request will be asserted until the system has performed the correct number of write operations.



.

Figure 13-20. Buffer Almost Empty Level (AEL)

13.3.2.10.4 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user should configure the MCSPI_XFERLEVEL register, the AEL and AFL levels, and, especially, the WCNT bit field to define the number of SPI word to be transferred using the FIFO. This should be done before enabling the channel.

This counter allows the controller to stop the transfer correctly after a defined number of SPI words have been transferred. If WNCT is cleared to 0, the counter is not used and the user must stop the transfer manually by disabling the channel, in this case the user doesn't know how many SPI transfers have been done. For receive transfer, software shall poll the corresponding FFE bit field and read the Receive register to empty the FIFO buffer.

When End Of Word count interrupt is generated, the user can disable the channel and poll on MCSPI_CH(i)STAT[FFE] register to know if SPI word is still there in FIFO buffer and read last words.

13.3.2.10.5 Multiple SPI Word Access

The CPU has the ability to perform multiple SPI word access to the receive or transmit registers within a single 32-bit OCP access by setting the bit field MCSPI_MODULCTRL[MOA] to '1' under specific conditions:

- The channel selected has the FIFO enable.
- Only FIFO sense enabled support the kind of access.
- The bit field MCSPI_MODULCTRL[MOA] is set to 1
- Only 32-bit OCP access and data width can be performed to receive or transmit registers, for other kind of access the CPU must de-assert MCSPI_MODULCTRL[MOA] bit fields.
- The Level MCSPI_XFERLEVEL[AEL] and MCSPI_XFERLEVEL[AFL] must be 32-bit aligned , it means that AEL[0] = AEL[1] = 1 or AFL[0] = AFL[1] = 1.
- If MCSPI_XFERLEVEL[WCNT] is used it must be configured according to SPI word length.
- The word length of SPI words allows to perform multiple SPI access, that means that MCSPI_CH(i)CONF[WL] < 16.

Number of SPI word access depending on SPI word length:

- 3 ≤ WL ≤ 7, SPI word length smaller or equal to byte length, four SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0]=WCNT[1]=0.
- 8 ≤ WL ≤ 15, SPI word length greater than byte or equal to 16-bit length, two SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0]= 0.
- 16 ≤ WL multiple SPI word access not applicable.

13.3.2.11 First SPI Word Delayed

The McSPI controller has the ability to delay the first SPI word transfer to give time for system to complete some parallel processes or fill the FIFO in order to improve transfer bandwidth. This delay is applied only on first SPI word after SPI channel enabled and first write in Transmit register. It is based on output clock frequency.

This option is meaningful in controller mode and single channel mode, MCSPI_MODULCTRL[SINGLE] = 1.

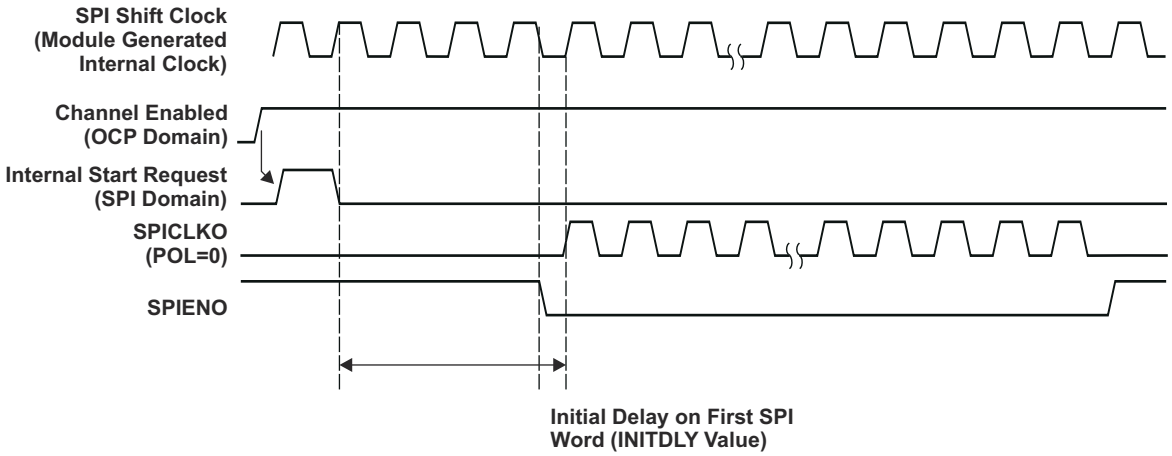


Figure 13-21. Controller Single Channel Initial Delay

Few delay values are available: No delay, 4/8/16/32 SPI cycles.

Its accuracy is half cycle in clock bypass mode and depends on clock polarity and phase.

13.3.2.12 3- or 4-Pin Mode

External SPI bus interface can be configured to use a restricted set of pins using the bit field MCSPI_MODULCTRL[PIN34] and depending on targeted application:

- If MCSPI_MODULCTRL[PIN34] is cleared to 0 (default value) the controller is in 4-pin mode using the SPI pins SPICLK, SOMI, SIMO and chip enable CS.
- If MCSPI_MODULCTRL[PIN34] is set to 1 the controller is in 3-pin mode using the SPI pins SPICLK, SOMI and SIMO.

In 3-pin mode it is mandatory to put the controller in single channel controller mode (MCSPI_MODULCTRL[SINGLE] asserted) and to connect only one SPI device on the bus.

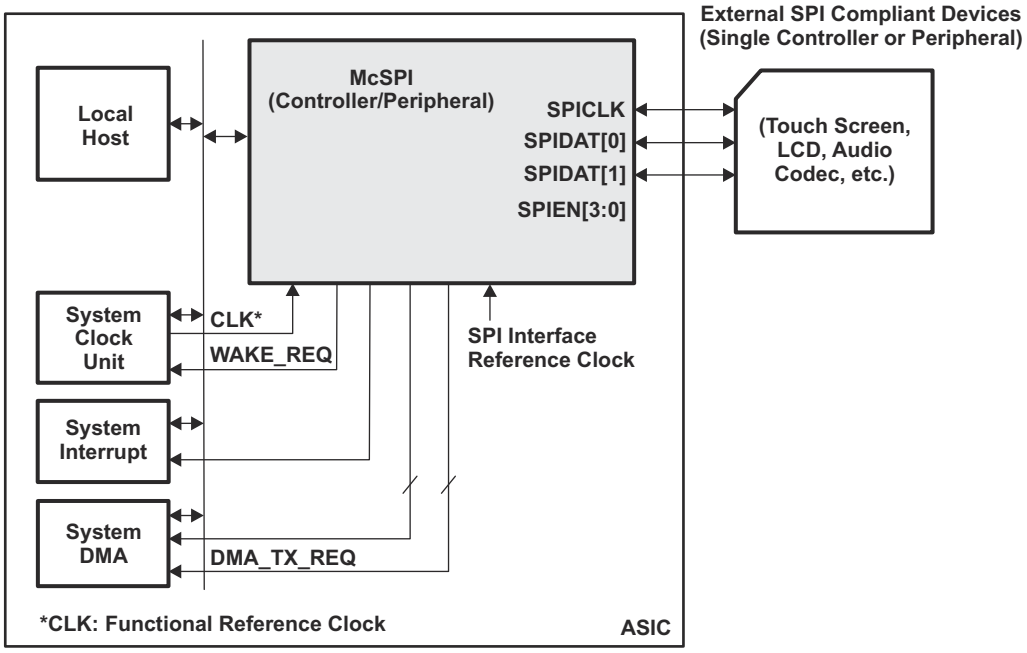


Figure 13-22. 3-Pin Mode System Overview

In 3-pin mode all options related to chip select management are useless:

- MCSPI_CHxCONF[EPOL]
- MCSPI_CHxCONF[TCS0]
- MCSPI_CHxCONF[FORCE]

The chip select pin SPIEN is forced to ‘0’ in this mode.

ADVANCE INFORMATION

13.3.3 Peripheral Mode

McSPI is in peripheral mode when the bit MS of the register MCSPI_MODULCTRL is set.

In peripheral mode, McSPI can be connected to up to 4 external SPI controller devices. McSPI handles transactions with a single SPI controller device at a time.

In peripheral mode, McSPI initiates data transfer on the data lines (SPIDAT[1;0]) when it receives an SPI clock (SPICLK) from the external SPI controller device.

The controller is able to work with or without a chip select SPIEN depending on MCSPI_MODULCTRL[PIN34] bit setting. It also supports transfers without a dead cycle between two successive words.

13.3.3.1 Dedicated Resources

In peripheral mode, enabling a channel that is not channel 0 has no effect. Only channel 0 can be enabled. The channel 0, in peripheral mode has the following resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH0CTRL. This channel should be enabled before transmission and reception. Disabling the channel, outside data word transmission, remains under user responsibility.
- Any of the 4 ports SPIEN[3:0] can be used as a peripheral SPI device enable. This is programmable with the bits SPIENSLV of the register MCSPI_CH0CONF.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH0STAT is set. When McSPI is selected by an external controller (active signal on the SPIEN port assigned to channel 0), the transmitter register content of channel0 is always loaded in shift register whether it has been updated or not. The transmitter register should be loaded before McSPI is selected by a controller.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH0STAT is set.

Note

The transmitter register and receiver registers of the other channels are not used. Read from or Write in the registers of a channel other than 0 has no effect.

- Its own communication configuration with the following parameters via the register MCSPI_CH0CONF:
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - SPIEN polarity, programmable with the bit EPOL.
  - SPICLK polarity, programmable with the bit POL.
  - SPICLK phase, programmable with the bit PHA.
  - Use a FIFO buffer or not, programmable with FFER and FFEW, depending on transfer mode TRM.

The SPICLK frequency of a transfer is controlled by the external SPI controller connected to McSPI. The bits CLKD0 of the MCSPI_CH0CONF register are not used in peripheral mode.

Note

The configuration of the channel can be loaded in the MCSPI_CH0CONF register only when the channel is disabled.

- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW of the MCSPI_CH0CONF register.
- Four interrupts events.

Figure 13-23 shows an example of four peripherals wired on a single controller device.

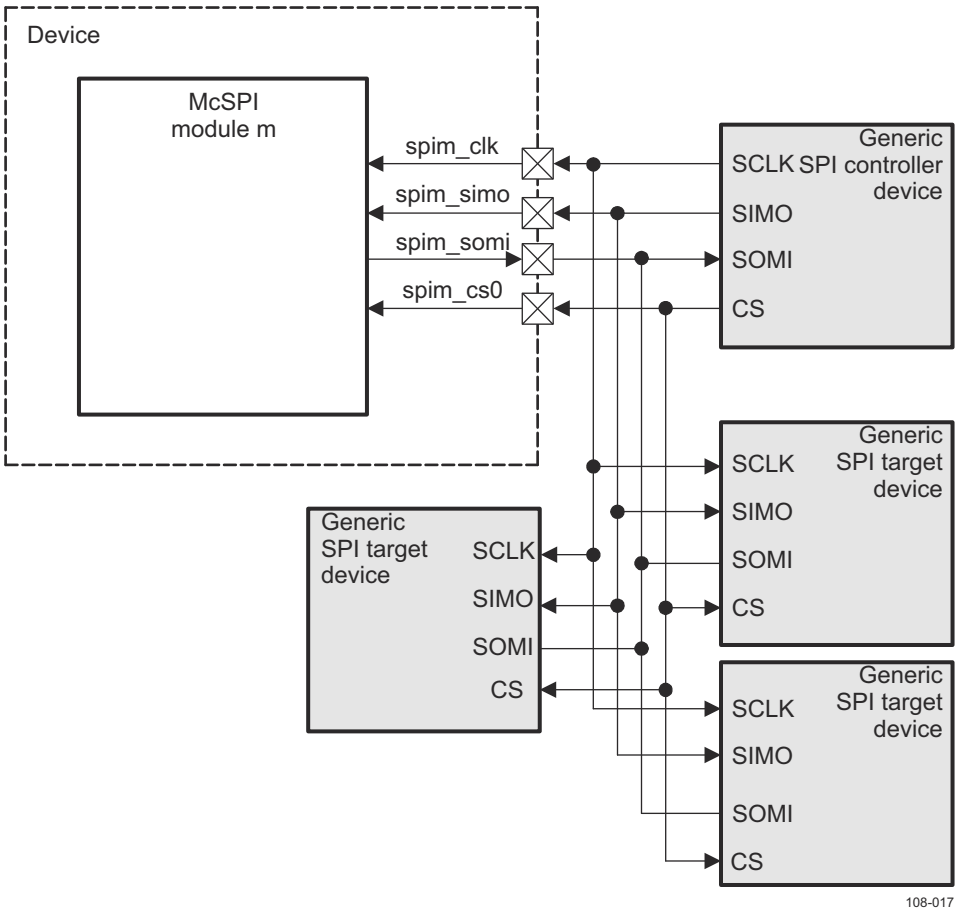


Figure 13-23. Example of SPI Peripheral with One Controller and Multiple Peripheral Devices on Channel 0

13.3.3.2 Interrupt Events in Peripheral Mode

The interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt events related to the receiver register state are RX_full and RX_overflow.

13.3.3.2.1 TX_EMPTY

The event TX_empty is activated when the channel is enabled and its transmitter register becomes empty. Enabling channel automatically raises this event. When FIFO buffer is enabled (MCSPI_CH(i)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in buffer to write a number of byte defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be load to remove source of interrupt and TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new TX_empty event will be asserted unless the host performs the number of writes to the transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of the Local Host to perform the right number of writes.

13.3.3.2.2 TX_UNDERFLOW

The event TX_underflow is activated when channel is enabled and if the transmitter register or FIFO (if use of buffer is enabled) is empty (not updated with new data) when an external controller device starts a data transfer with McSPI (transmit and receive).

When the FIFO is enabled, the data read while the underflow flag is set will not be the last word written to the FIFO.

The TX_underflow indicates an error (data loss) in peripheral mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

13.3.3.2.3 RX_FULL

The event RX_FULL is activated when channel is enabled and receiver becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(i)CONF[FFER] set to 1), the RX_FULL is asserted as soon as there is a number of bytes holds in buffer to read defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new RX_FULL event will be asserted unless the host has performed the number of reads from the receive register defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of Local Host to perform the right number of reads.

13.3.3.2.4 RX_OVERFLOW

The RX0_OVERFLOW event is activated in peripheral mode in either transmit-and-receive or receive-only mode, when a channel is enabled and the SPI_RX*n* register or FIFO is full when a new SPI word is received. The SPI_RX*n* register is always overwritten with the new SPI word. If the FIFO is enabled, data within the FIFO is overwritten, it must be considered as corrupted. The RX0_OVERFLOW event should not appear in peripheral mode using the FIFO.

The RX0_OVERFLOW indicates an error (data loss) in peripheral mode.

The SPI_IRQSTATUS[3] RX0_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

13.3.3.2.5 End of Word Count

The event end of word (EOW) count is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The EOW count interrupt also indicates that the SPI transfer has halted on the channel using the FIFO buffer.

The EOW interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

13.3.3.3 Peripheral Transmit-and-Receive Mode

The peripheral transmit and receive mode is programmable (TRM bit cleared to 0 in the register MCSPI_CH(i)CONF).

After the channel is enabled, transmission and reception proceeds with interrupt and DMA request events.

In peripheral transmit and receive mode, transmitter register should be loaded before McSPI is selected by an external SPI controller device.

Transmitter register or FIFO (if enabled) content is always loaded into the shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured in one data direction, transmit or receive, then the FIFO is seen as a unique 64-byte buffer. It can also be configured in both data directions, transmit and receive, then the FIFO is split into two separate 32-byte buffers with their own address space management.

13.3.3.4 Peripheral Receive-Only Mode

The peripheral receive-only mode is programmable (MCSPI_CH(i)CONF[TRM] set to 01).

In receive-only mode, the transmitter register should be loaded before McSPI is selected by an external SPI controller device. Transmitter register or FIFO (if enabled) content is always loaded into the shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

When an SPI word transfer completes (the MCSPI_CH(i)STAT[EOT] bit (with *I* = 0) is set to 1), the received data is transferred to the channel receive register.

To use McSPI as a peripheral receive-only device with MCSPI_CH(i)CONF[TRM]=00, the user has the responsibility to disable the TX_empty and TX_underflow interrupts and DMA write requests due to the transmitter register state.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(i)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel. The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

Figure 13-24 shows an example of a half-duplex system with a controller device on the left and a receive-only peripheral device on the right. Each time one bit transfers out from the controller, one bit transfers in to the peripheral. If WordA is 8 bits, then after eight cycles of the serial clock spim_clk, WordA transfers from the controller to the peripheral.

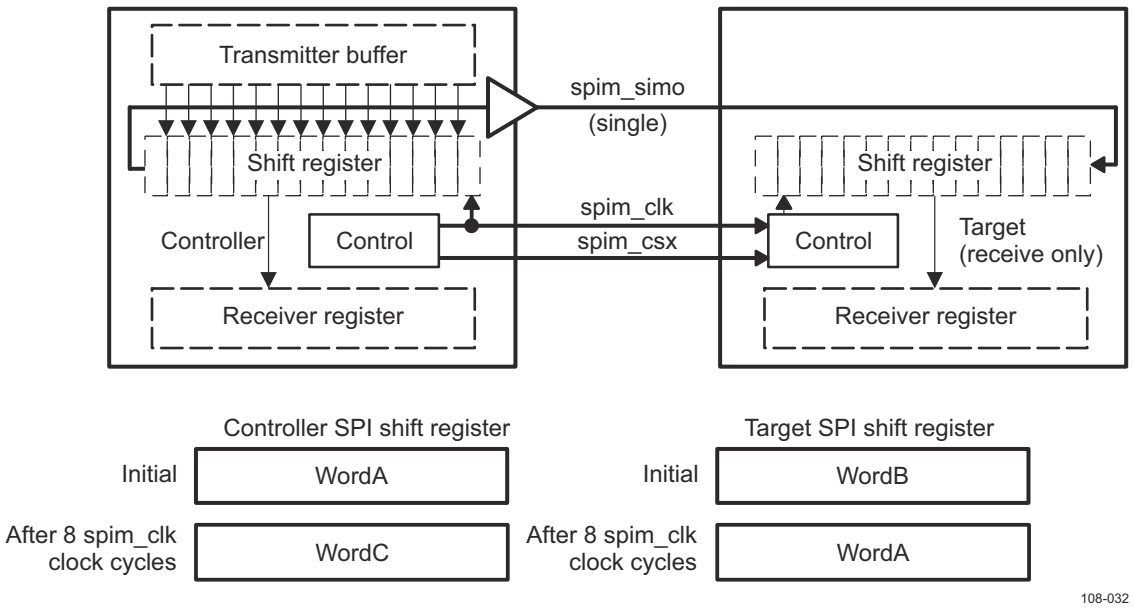


Figure 13-24. SPI Half-Duplex Transmission (Receive-Only Peripheral)

13.3.3.5 Peripheral Transmit-Only Mode

The peripheral transmit-only mode is programmable (MCSPI_CH(i)CONF[TRM] set to 10). This mode eliminates the need for the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

To use McSPI as a peripheral transmit-only device with MCSPI_CH(i)CONF[TRM]=10, the user should disable the RX_full and RX_overflow interrupts and DMA read requests due to the receiver register state.

On completion of SPI word transfer the bit EOT of the register MCSPI_CH(i)STAT is set. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(i)CONF register, then the FIFO is seen as a unique 64-byte buffer.

Figure 13-25 shows a half-duplex system with a controller device on the left and a transmit-only peripheral device on the right. Each time a bit transfers out from the peripheral device, one bit transfers in the controller. If WordB is 8-bits, then after eight cycles of the serial clock spim_clk, WordB transfers from the peripheral to the controller.

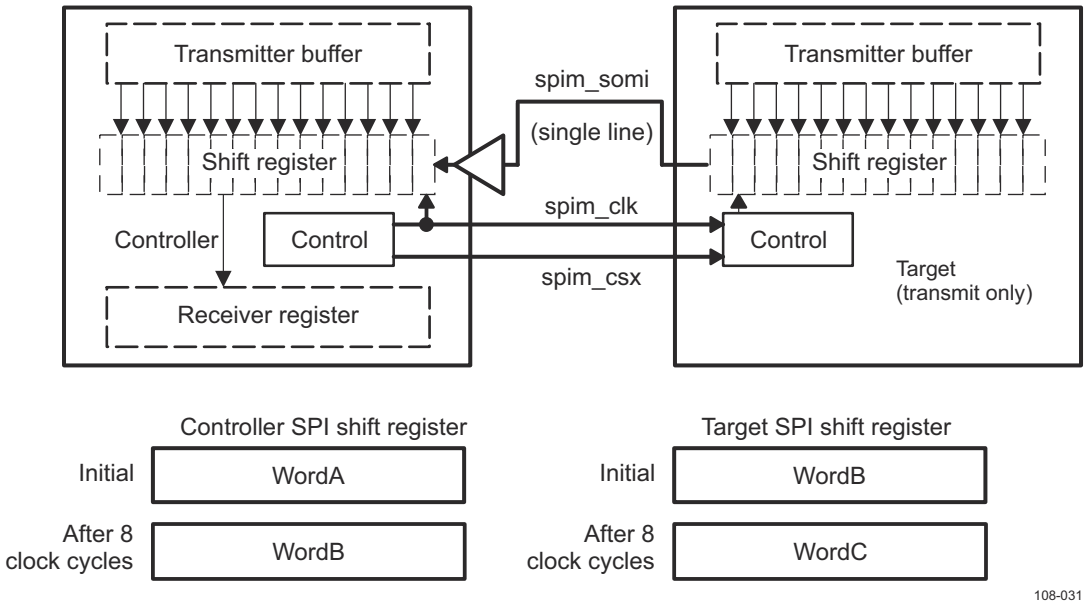


Figure 13-25. SPI Half-Duplex Transmission (Transmit-Only Peripheral)

13.3.4 Interrupts

According to its transmitter register state and its receiver register state each channel can issue interrupt events if they are enabled.

The interrupt events are listed in the Section 13.3.2.2 and in Section 13.3.3.2.

Each interrupt event has a status bit, in the MCSPI_IRQSTATUS register, which indicates service is required, and an interrupt enable bit, in the MCSPI_IRQENABLE register, which enables the status to generate hardware interrupt requests.

When an interrupt occurs and it is later masked (IRQENABLE), the interrupt line is not asserted again even if the interrupt source has not been serviced.

McSPI supports interrupt driven operation and polling.

13.3.4.1 Interrupt-Driven Operation

Alternatively, an interrupt enable bit, in the MCSPI_IRQENABLE register, can be set to enable each of the events to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the CPU must:

- Read the MCSPI_IRQSTATUS register to identify which event occurred,
- Read the receiver register that corresponds to the event in order to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event in order to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Write a 1 into the corresponding bit of MCSPI_IRQSTATUS register to clear the interrupt status, and release the interrupt line.

The interrupt status bit should always be reset after the channel is enabled and before the event is enabled as an interrupt source.

13.3.4.2 Polling

When the interrupt capability of an event is disabled in the MCSPI_IRQENABLE register, the interrupt line is not asserted and:

- The status bits in the MCSPI_IRQSTATUS register can be polled by software to detect when the corresponding event occurs.
- Once the expected event occurs, CPU must read the receiver register that corresponds to the event in order to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event in order to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Writing a 1 into the corresponding bit of MCSPI_IRQSTATUS register clears the interrupt status and does not affect the interrupt line state.

13.3.5 DMA Requests

McSPI can be interfaced with a DMA controller. At system level, the advantage is to free the local host of the data transfers.

According to its transmitter register state, its receiver register state or FIFO level (if use of buffer for the channel) each channel can issue DMA requests if they are enabled.

The DMA requests need to be disabled in order to get TX and RX interrupts, in order to define either the end of the transfer or the transfer of the last words for the modes listed below:

- Controller transmit-only
- Controller normal receive-only mode
- Controller turbo receive-only mode
- Peripheral transmit-only

There are two DMA request lines per channel. The management of DMA requests differ according to use of FIFO buffer or not.

13.3.5.1 FIFO Buffer Disabled

The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(i)CONF. The DMA Read request line is de-asserted on read completion of the receive register of the channel.

The DMA Write request line is asserted when the channel is enabled and the transmitter register of the channel is empty. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(i)CONF. The DMA Write request line is de-asserted on load completion of the transmitter register of the channel.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive register.

13.3.5.2 FIFO Buffer Enabled

The DMA Read request line is asserted when the channel is enabled and a number of bytes defined in MCSPI_XFERLEVEL[AFL] bit field is hold in FIFO buffer for the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(i)CONF. The DMA Read request line is de-asserted on the first SPI word read completion of the receive register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of read accesses defined by MCSPI_XFERLEVEL[AFL] it is under user responsibility.

The DMA Write request line is asserted when the channel is enabled and the number of bytes hold in FIFO buffer is below the level defined by the MCSPI_XFERLEVEL[AEL] bit field. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(i)CONF. The DMA Write request line is de-asserted on load completion of the first SPI word in the transmitter register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of write accesses defined by MCSPI_XFERLEVEL[AEL] it is under user responsibility.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive FIFO.

13.3.5.3 DMA 256-Bit Aligned Addresses

The controller has two registers, MCSPI_DAFTX and MCSPI_DAFRX, used only with an enabled channel which manages the FIFO to be compliant the a DMA handler providing only 256-bit aligned addresses.

This features is activated when the bit field MCSPI_MODULCTRL[FDDA] is set to ‘1’ and only one enabled channel have its bit field MCSPI_CH(i)CONF[FFEW] or MCSPI_CH(i)CONF[FFER] enabled.

In this case the registers MCSPI_TX(i) and MCSPI_RX(i) are not used and data is managed through registers MCSPI_DAFTX and MCSPI_DAFRX.

13.3.6 Emulation Mode

The MReqDebug input differentiates a regular access of a processor (application access), from an emulator access.

Application access: MReqDebug = 0

In functional mode, the consequences of a read of a receiver register MCSPI_RX(i) are the following:

- The source of an RX(i)_Full event in the MCSPI_IRQSTATUS register is removed, if it was enabled in the MCSPI_IRQENABLE register.
- The RX(i)S status bit in the MCSPI_IRQSTATUS register is cleared.
- In controller mode, depending on the round robin arbitration, and the transmitter register state, the channel may access to the shift register for transmission/reception.

Emulator access: MReqDebug = 1

In emulation mode, McSPI behavior is the same as in functional mode but a read of a receiver register MCSPI_RX(i) is not intrusive:

- MCSPI_RX(i) is still considered as not read. When the FIFO buffer is enabled, pointers are not updated.
- The source of an RX(i)_Full event in the MCSPI_IRQSTATUS register is not removed. The RX(i)S status bit in the MCSPI_CH(i)STAT register is held steady.

In emulation mode, as in functional mode, based on the ongoing data transfers, the status bits of the MCSPI_CH(i)STAT register may be optionally updated, the interrupt and DMA request lines may be optionally asserted.

13.3.7 Power Saving Management

Independently of the module operational modes (Transmit and/or Receive), two modes of operations are defined from a power management perspective: normal and idle modes.

The two modes are temporally fully exclusive.

13.3.7.1 Normal Mode

Both the Interface, or OCP, clock and SPI clock (CLKSPIREF) provided to McSPI must be active for both controller and peripheral modes. The auto-gating of the module OCP clock and SPI clock occurs when the following conditions are met:

- The bit AutoIdle of the register MCSPI_SYSCONFIG is set.
- In controller mode, there is no data to transmit or receive in all channels.
- In peripheral mode, the SPI is not selected by the external SPI controller device and no OCP accesses.

Autogating of the module OCP clock and SPI clock stops when the following conditions are met:

- In controller mode, an OCP access occurs.
- In peripheral mode, an OCP access occurs or McSPI is selected by an external SPI controller device.

13.3.7.2 Idle Mode

The OCP clock and SPI clock provided to McSPI may be switched off on system power manager request and switched back on module request.

McSPI is compliant with the power management handshaking protocol: idle request from the system power manager, idle acknowledgement from McSPI.

The idle acknowledgement in response to an idle request from the system power manager varies according to a programmable mode in the MCSPI_SYSCONFIG register: No idle mode, force idle mode, and smart idle mode.

- When programmed for no idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “01”), the module ignores the system power manager request, and behaves normally, as if the request was not asserted.
- When programmed for smart idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “10”), the module acknowledges the system power manager request according to its internal state.
- When programmed for force idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “00”), the module acknowledges the system power manager request unconditionally.

The OCP clock will be optionally switched off, during the smart idle mode period, if the bit ClockActivity of the register MCSPI_SYSCONFIG is set.

The SPI clock will be optionally switched off, during the smart idle mode period, if the second bit ClockActivity of the register MCSPI_SYSCONFIG is set.

McSPI assumes that both clocks may be switched off whatever the value set in the field ClockActivity of the register MCSPI_SYSCONFIG.

13.3.7.2.1 Transitions from Normal Mode to Smart-Idle Mode

The module detects an idle request when the synchronous signal IdleReq is asserted.

When IdleReq is asserted, any access to the module will generate an error as long as the OCP clock is alive.

When configured as a peripheral device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of the current transfer (SPIEN peripheral selection signal deasserted by the external controller) and if interrupt or DMA request lines are not asserted.

As a controller device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of all the channel data transfers and if interrupt or DMA request lines are not asserted.

As long as SIdleAck is not asserted, if an event occurs, the module can still generate an interrupt or a DMA request after IdleReq assertion. In this case, the module ignores the idle request and SIdleAck will not get asserted: The system power manager will abort the power mode transition procedure. It is then the responsibility of the system to de-assert IdleReq before attempting to access the module.

When SIdleAck is asserted, the module does not assert any new interrupt or DMA request.

13.3.7.2.2 Transition From Smart-Idle Mode to Normal Mode

McSPI detects the end of the idle period when the idle request signal (IdleReq) is deasserted.

Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational.

13.3.7.2.3 Force-Idle Mode

Force-idle mode is enabled as follows:

- The bit SIdleMode of the register MCSPI_SYSCONFIG is cleared to “00” (Force Idle).

  The force idle mode is an idle mode where McSPI responds unconditionally to the idle request by asserting the SIdleAck signal and by deasserting unconditionally the interrupt and DMA request lines if asserted.

  The transition from normal mode to idle mode does not affect the interrupt event bits of the MCSPI_IRQSTATUS register.

  In force-idle mode, the module is supposed to be disabled at that time, so the interrupt and DMA request lines are likely deasserted. OCP clock and SPI clock provided to McSPI can be switched off.

  An idle request during an SPI data transfer can lead to an unexpected and unpredictable result, and is under software responsibility.

  Any access to the module in force idle mode will generate an error as long as the OCP clock is alive and IdleReq is asserted.

The module exits the force idle mode when:

- The idle request signal (IdleReq) is de-asserted.

  Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational. The interrupt and DMA request lines are optionally asserted a clock cycle later.

13.3.8 System Test Mode

McSPI is in system test mode (SYSTEST) when the bit System_Test of the register MCSPI_MODULCTRL is set.

The SYSTEST mode is used to check in a very simple manner the correctness of the system interconnect either internally to interrupt handler, or power manager, or externally to SPI I/Os.

I/O verification can be performed in SYSTEST mode by toggling the outputs and capturing the logic state of the inputs. (See MCSPI_SYST register definition in Section 13.7)

13.3.9 Reset

13.3.9.1 Internal Reset Monitoring

The module is reset by the hardware when an active-low reset signal, synchronous to the OCP interface clock is asserted on the input pin RESETN.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset, in all clock domains.

Additionally, the module can be reset by software through the bit SoftReset of the register MCSPI_SYSCONFIG. This bit has exactly the same action on the module logic as the hardware RESETN signal. The register MCSPI_SYSCONFIG is not sensitive to software reset. The SoftReset control bit is active high. The bit is automatically reset to 0 by the hardware.

A global ResetDone status bit is provided in the status register MCSPI_SYSSTATUS. This bit is set to 1 when all the different clock domains resets (OCP domain and SPI domains) have been released (logical AND).

The global ResetDone status bit can be monitored by the software to check if the module is ready-to-use following a reset (either hardware or software).

The clock CLKSPIREF must be provided to the module, in order to allow the ResetDone status bit to be set.

When used in peripheral mode, the clock CLKSPIREF is needed only during the reset phase. The clock CLKSPIREF can be switched off after the ResetDone status is set.

13.3.10 Access to Data Registers

This section details the supported data accesses (read or write) from/to the data receiver registers MCSPI_RX(i) and data transmitter registers MCSPI_TX(i).

Supported access:

McSPI supports only one SPI word per register (receiver or transmitter) and does not support successive 8-bit or 16-bit accesses for a single SPI word.

The SPI word received is always right justified on LSbit of the 32bit register MCSPI_RX(i), and the SPI word to transmit is always right justified on LSbit of the 32bit register MCSPI_TX(i).

The upper bits, above SPI word length, are ignored and the content of the data registers is not reset between the SPI data transfers.

The coherence between the number of bits of the SPI Word, the number of bits of the access and the enabled byte remains under the user's responsibility. Only aligned accesses are supported.

In Controller mode, data should not be written in the transmit register when the channel is disabled.

13.3.11 Programming Aid

13.3.11.1 Module Initialization

- Hard or soft reset.
- Read MCSPI_SYSSTATUS.
- Check if reset is done.
- Module configuration: (a) Write into MCSPI_MODULCTRL (b) Write into MCSPI_SYSCONFIG.
- Before the ResetDone bit is set, the clocks CLK and CLKSPIREF must be provided to the module.
- To avoid hazardous behavior, it is advised to reset the module before changing from CONTROLLER mode to PERIPHERAL mode or from PERIPHERAL mode to CONTROLLER mode.

13.3.11.2 Common Transfer Sequence

McSPI module allows the transfer of one or several words, according to different modes:

- CONTROLLER, CONTROLLER Turbo, PERIPHERAL
- TRANSMIT - RECEIVE, TRANSMIT ONLY, RECEIVE ONLY
- Write and Read requests: Interrupts, DMA
- SPIEN lines assertion/deassertion: automatic, manual

For all these flows, the host process contains the main process and the interrupt routines. The interrupt routines are called on the interrupt signals or by an internal call if the module is used in polling mode.

In multi-channel controller mode, the flows of different channels can be run simultaneously.

13.3.11.3 Main Program

- Interrupt Initialization: (a) Reset status bits in MCSPI_IRQSTATUS (b) Enable interrupts in MCSPI_IRQENA.
- Channel Configuration: Write MCSPI_CH(i)CONF.
- Start the channel: Write 0000 0001h in MCSPI_CH(i)CTRL.
- First write request: TX empty - Generate DMA write event/ polling TX empty flag by CPU to write First transmit word into MCSPI_TX(i).
- End of transfer: Stop the channel by writing 0000 0000h in MCSPI_CH(i)CTRL

The end of transfer depends on the transfer mode.

In multi-channel controller mode, be careful not to overwrite the bits of other channels when initializing MCSPI_IRQSTATUS and MCSPI_IRQENABLE.

13.3.12 Interrupt and DMA Events

McSPI has two DMA requests (Rx and Tx) per channel. It also has one interrupt line for all the interrupt requests.

13.4 Frequency of Operation

Recommended frequency of operation based on timing closure.

	SPI1/A	SPI2/B
Master TX Only	40MHz	54MHz
RX / Bidirectional	40MHz	40MHz
IO loopback	Supported	Supported

- One pin mux mode of SPIB which corresponds to PAD_AW clock is met at lower freq (20MHz for master and 20M for slave). (Captured in RIOT sheet v0p6).

13.5 McSPI Smart Idle Implementation

The following diagram describes the Smart Idle Implementation for McSPI.

Device supports two modes of Smart IDLE.

- Manual Mode - In this mode of operation, SMART_IDLE_WAKE_AUTO_EN = 0. The control bit is directly connected to the SPI_CLKSTOP_REQ. The entry and exit to Smart Idle is user controlled based on polling SMART_IDLE_ACK and SMART_IDLE_WAKEUP.
- Automatic Mode - In this mode of operation, entry to smart idle mode is manual by setting SMART_IDLE_ENABLE = 1. When the clkstop_wakeup signal from McSPI is asserted (based on the activity), the clkstop_req is pulled low automatically.

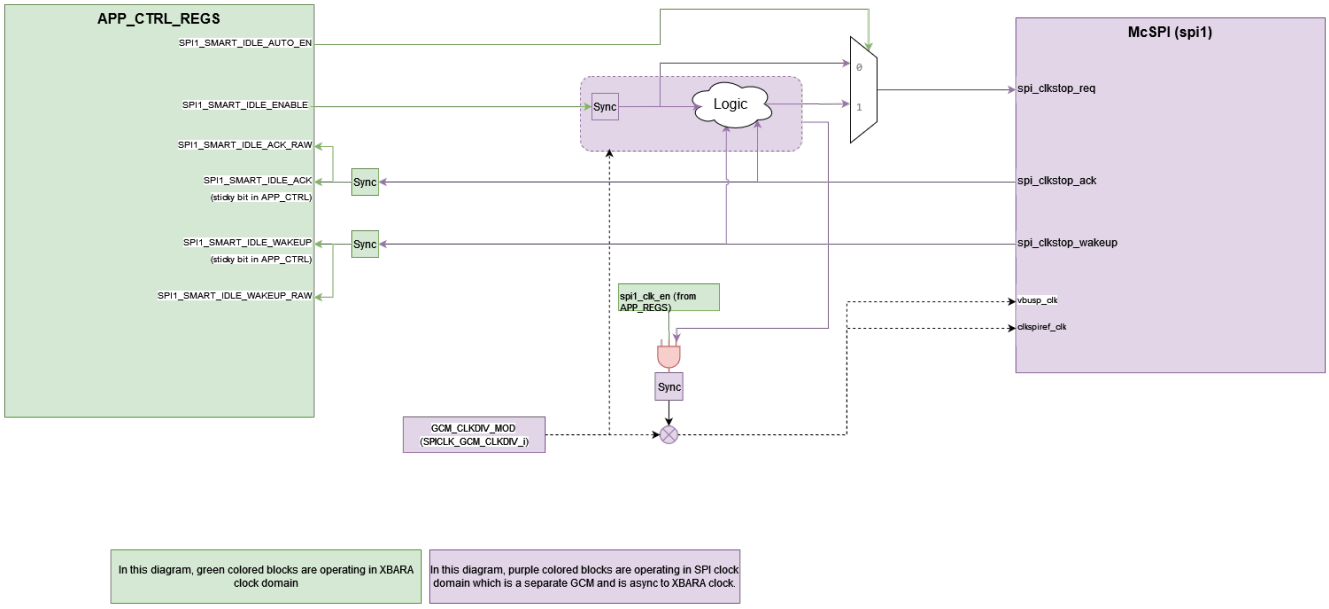


Figure 13-26. mcSPI CLKSTOP Logic

The table below shows the truth table of the logic shown in cloud in the above diagram.

SMART_IDLE_ENABLE	CLKSTOP_WAKEUP	cLKSTOP_ACK	CLKSTOP_REQ(AUTO_WAKEUP=1)	ICG_EN(AUTO_WAKEUP =1/0)
0	0	0	Old State	1
0	0	1	Old State	1
0	1 (valid is slave mode)	0	Old State	1
0	1 (valid is slave mode)	1	Old State	1
1	0	0	1	1
1	**0**	**1**	**1**	**0 = clk shut**
1	1 (valid is slave mode)	0	0	1

SMART_IDLE_ENABLE	CLKSTOP_WAKEUP	cLKSTOP_ACK	CLKSTOP_REQ(AUTO_WAKEUP=1)	ICG_EN(AUTO_WAKEUP=1/0)
1	1 (valid is slave mode)	1	0	1

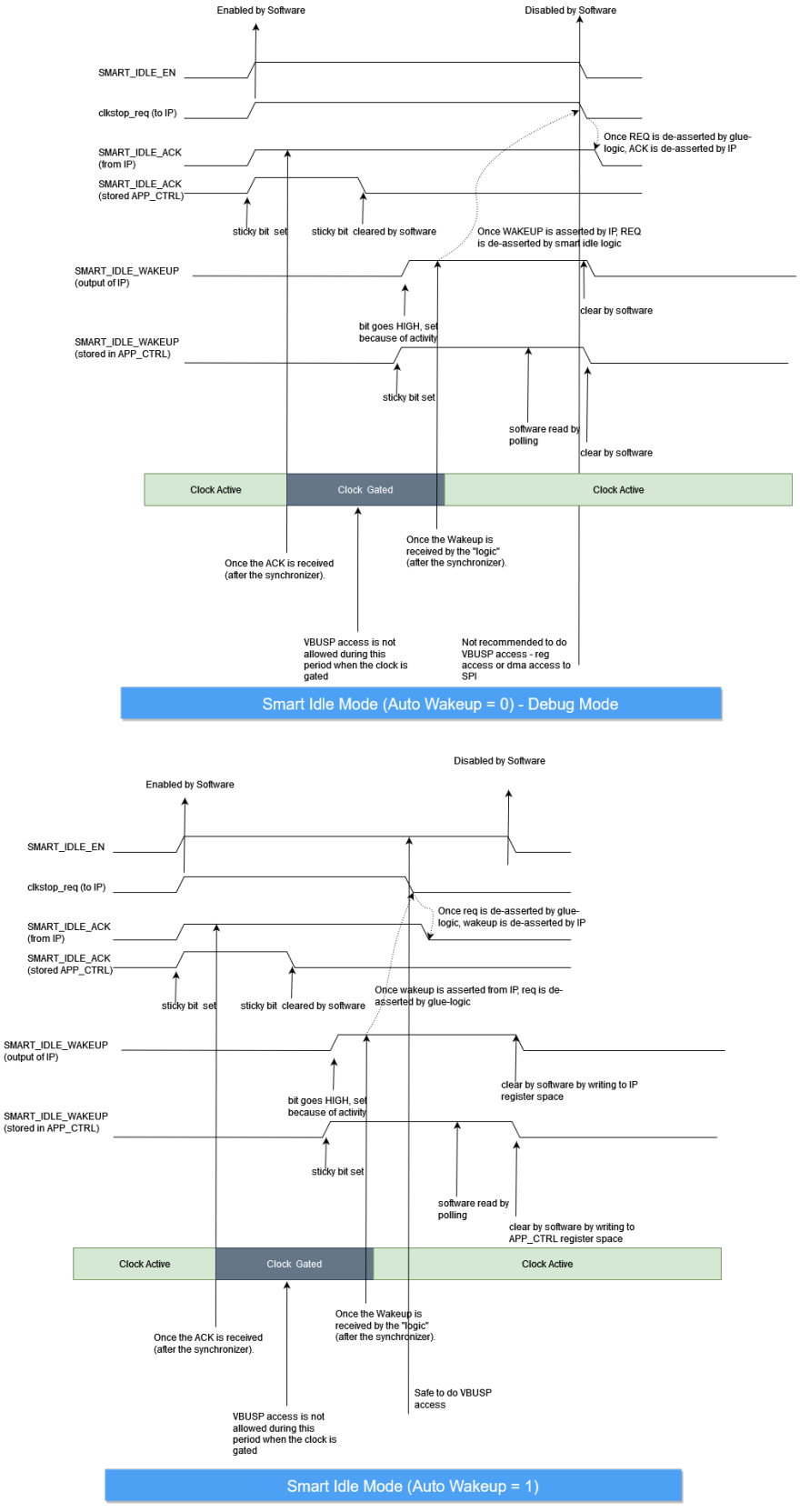


Figure 13-27. mcSPI CLKSTOP Waveform

13.6 Programming Sequence for Smart IDLE

Auto Wakeup = 1 & Master mode

1. Configure McSPI as required
2. Enable SmartIdle (by setting SPI1_SMART_IDLE_ENABLE=1) after ensuring that there is **no** pending transaction from/to SPI or any more access to be done to McSPI by CPU or DMA
3. If any register or memory access to McSPI has to be done, disable SmartIDLE mode (by setting SPI1_SMART_IDLE_ENABLE=0)
4. In Master mode, the external host is not going to toggle the SPI_CS, hence there will not be any wakeup => there is no difference between AUTO_WAKEUP=1 or 0.

Auto Wakeup = 1 & Slave mode

1. Configure McSPI as required
2. Enable SmartIdle (by setting SPI1_SMART_IDLE_ENABLE=1) after ensuring that there is **no** pending transaction from/to SPI or any more access to be done to McSPI by CPU or DMA
3. If any register or memory access to McSPI has to be done by any master (DMA / CPU), disable SmartIDLE mode (by setting SPI1_SMART_IDLE_ENABLE=0)
4. If there is wakeup from McSPI (because of some SPI_CS toggle), then the clock is automatically enabled.
5. Disable SmartIdle configuration (by setting SPI1_SMART_IDLE_ENABLE=0) - to do the register access.

13.7 McSPI Registers

13.7.1 APP_SPI Registers

Table 13-10 lists the memory-mapped registers for the APP_SPI registers. All register offset addresses not listed in Table 13-10 should be considered as reserved locations and the register contents should not be modified.

Table 13-10. APP_SPI Registers

Offset	Acronym	Register Name	Section
0h	HL_REV	HL_REV	Go
4h	HL_HWINFO	HL_HWINFO	Go
10h	HL_SYSCONFIG	HL_SYSCONFIG	Go
100h	REVISION	REVISION	Go
110h	SYSCONFIG	SYSCONFIG	Go
114h	SYSSTATUS	SYSSTATUS	Go
118h	IRQSTATUS	IRQSTATUS	Go
11Ch	IRQENABLE	IRQENABLE	Go
120h	WAKEUPENABLE	WAKEUPENABLE	Go
124h	SYST	SYST	Go
128h	MODULCTRL	MODULCTRL	Go
12Ch	CH0CONF	CH0CONF	Go
130h	CH0STAT	CH0STAT	Go
134h	CH0CTRL	CH0CTRL	Go
138h	TX0	TX0	Go
13Ch	RX0	RX0	Go
140h	CH1CONF	CH1CONF	Go
144h	CH1STAT	CH1STAT	Go
148h	CH1CTRL	CH1CTRL	Go
14Ch	TX1	TX1	Go
150h	RX1	RX1	Go
154h	CH2CONF	CH2CONF	Go
158h	CH2STAT	CH2STAT	Go
15Ch	CH2CTRL	CH2CTRL	Go
160h	TX2	TX2	Go
164h	RX2	RX2	Go
168h	CH3CONF	CH3CONF	Go
16Ch	CH3STAT	CH3STAT	Go
170h	CH3CTRL	CH3CTRL	Go
174h	TX3	TX3	Go
178h	RX3	RX3	Go
17Ch	XFERLEVEL	XFERLEVEL	Go
180h	DAFTX	DAFTX	Go
1A0h	DAFRX	DAFRX	Go

Complex bit access types are encoded to fit into small table cells. Table 13-11 shows the codes that are used for access types in this section.

Table 13-11. APP_SPI Access Type Codes

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

Table 13-11. APP_SPI Access Type Codes (continued)

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

13.7.1.1 HL_REV Register (Offset = 0h) [Reset = 40301A0Bh]

HL_REV is shown in Table 13-12.

Return to the Summary Table.

IP Revision Identifier (X.Y.R) Used by software to track features bugs and compatibility

Table 13-12. HL_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Used to distinguish between old scheme and current - (RO)
29-28	RSVD	R	0h	Reserved These bits are initialized to zero and writes to them are ignored - (RO)
27-16	FUNC	R	30h	Function indicates a software compatible module family If there is no level of software compatibility a new Func number [and hence REVISION] should be assigned - (RO)
15-11	R_RTL	R	3h	RTL Version [R] maintained by IP design owner RTL follows a numbering such as XYRZ which are explained in this table R changes ONLY when: [1] PDS uploads occur which may have been due to spec changes [2] Bug fixes occur [3] Resets to '0' when X or Y changes Design team has an internal 'Z' [customer invisible] number which increments on every drop that happens due to DV and RTL updates Z resets to 0 when R increments - (RO)
10-8	X_MAJOR	R	2h	Major Revision [X] maintained by IP specification owner X changes ONLY when: [1] There is a major feature addition An example would be adding Master Mode to Utopia Level2 The Func field [or Class/ Type in old PID format] will remain the same X does NOT change due to: [1] Bug fixes [2] Change in feature parameters - (RO)
7-6	CUSTOM	R	0h	Indicates a special version for a particular device Consequence of use may avoid use of standard Chip Support Library [CSL] / Drivers - (RO)
5-0	Y_MINOR	R	Bh	Minor Revision [Y] maintained by IP specification owner Y changes ONLY when: [1] Features are scaled [up or down] Flexibility exists in that this feature scalability may either be represented in the Y change or a specific register in the IP that indicates which features are exactly available [2] When feature creeps from Is-Not list to Is list But this may not be the case once it sees silicon in which case X will change Y does NOT change due to: [1] Bug fixes [2] Typos or clarifications [3] major functional/feature change/addition/deletion Instead these changes may be reflected via R S X as applicable Spec owner maintains a customer-invisible number 'S' which changes due to: [1] Typos/clarifications [2] Bug documentation Note that this bug is not due to a spec change but due to implementation Nevertheless the spec tracks the IP bugs An RTL release [say for silicon PG11] that occurs due to bug fix should document the corresponding spec number [XYS] in its release notes - (RO)

13.7.1.2 HL_HWINFO Register (Offset = 4h) [Reset = 00000021h]

HL_HWINFO is shown in Table 13-13.

Return to the Summary Table.

Information about the IP module's hardware configuration i.e. typically the module's HDL generics (if any). Actual field format and encoding is up to the module's designer to decide.

Table 13-13. HL_HWINFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RSVD	R	0h	Reserved These bits are initialized to zero and writes to them are ignored - (RO)
6	RETMODE	R	0h	This bit field indicates whether the retention mode is supported using the pin PIRFFRET - (RO)
5-1	FFNBYTE	R	10h	FIFO number of byte generic parameter This register defines the value of FFNBYTE generic parameter only MSB bits from 8 down to 4 are taken into account - (RO)
0	USEFIFO	R	1h	Use of a FIFO enable: This bit field indicates if a FIFO is integrated within controller design with its management - (RO)

13.7.1.3 HL_SYSCONFIG Register (Offset = 10h) [Reset = 00000008h]

HL_SYSCONFIG is shown in Table 13-14.

Return to the Summary Table.

Clock management configuration

Table 13-14. HL_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RSVD	R	0h	Reserved - (RO)
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 - (RW)
1	FREEEMU	R/W	0h	Sensitivity to emulation [debug] suspend input signal - (RW)
0	SOFTRESET	R/W	0h	Software reset [Optional] - (RW)

13.7.1.4 REVISION Register (Offset = 100h) [Reset = 0000002Bh]

REVISION is shown in Table 13-15.

Return to the Summary Table.

This register contains the hard coded RTL revision number.

Table 13-15. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reads returns 0 - (RO)
7-0	REV	R	2Bh	IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 10 0x21 for 21 - (RO

13.7.1.5 SYSCONFIG Register (Offset = 110h) [Reset = 00000015h]

SYSCONFIG is shown in Table 13-16.

Return to the Summary Table.

This register allows controlling various parameters of the OCP interface.

Table 13-16. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reads returns 0 - (RO)
9-8	CLOCKACTIVITY	R/W	0h	Clocks activity during wakeup mode period - (RW)
7-5	RESERVED	R	0h	Reads returns 0 - (RO)
4-3	SIDLEMODE	R/W	2h	Power management - (RW)
2	ENAWAKEUP	R/W	1h	WakeUp feature control - (RW)
1	SOFTRESET	R/W	0h	Software reset During reads it always returns 0 - (RW)
0	AUTOIDLE	R/W	1h	Internal OCP Clock gating strategy - (RW)

13.7.1.6 SYSSTATUS Register (Offset = 114h) [Reset = 00000001h]

SYSSTATUS is shown in Table 13-17.

Return to the Summary Table.

This register provides status information about the module excluding the interrupt status information

Table 13-17. SYSSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved for module specific status information Read returns 0 - (RO)
0	RESETDONE	R	1h	Internal Reset Monitoring - (RO)

13.7.1.7 IRQSTATUS Register (Offset = 118h) [Reset = 00000000h]

IRQSTATUS is shown in Table 13-18.

Return to the Summary Table.

The interrupt status regroups all the status of the module internal events that can generate an interrupt

Table 13-18. IRQSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reads returns 0 - (RO)
17	EOW	R/W	0h	End of word count event when a channel is enabled using the FIFO buffer and the channel had sent the number of SPI word defined by MCSPI_XFERLEVELWCNT - (RW)
16	WKS	R/W	0h	Wakeup event in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CH0CONFSPIENSLV - (RW)
15	RESERVED	R	0h	Reads returns 0 - (RO)
14	RX3_FULL	R/W	0h	Receiver register is full or almost full Only when Channel 3 is enabled - (RW)
13	TX3_UNDERFLOW	R/W	0h	Transmitter register underflow Only when Channel 3 is enabled The transmitter register is empty [not updated by Host or DMA with new data] before its time slot assignment Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled - (RW)
12	TX3_EMPTY	R/W	0h	Transmitter register is empty or almost empty Note: Enabling the channel automatically rises this event - (RW)
11	RESERVED	R	0h	Reads returns 0 - (RO)

Table 13-18. IRQSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RX2_FULL	R/W	0h	Receiver register full or almost full Channel 2 - (RW)
9	TX2_UNDERFLOW	R/W	0h	Transmitter register underflow Channel 2 - (RW)
8	TX2_EMPTY	R/W	0h	Transmitter register empty or almost empty Channel 2 - (RW)
7	RESERVED	R	0h	Reads returns 0 - (RO)
6	RX1_FULL	R/W	0h	Receiver register full or almost full Channel 1 - (RW)
5	TX1_UNDERFLOW	R/W	0h	Transmitter register underflow Channel 1 - (RW)
4	TX1_EMPTY	R/W	0h	Transmitter register empty or almost empty Channel 1 - (RW)
3	RX0_OVERFLOW	R/W	0h	Receiver register overflow [slave mode only] Channel 0 - (RW)
2	RX0_FULL	R/W	0h	Receiver register full or almost full Channel 0 - (RW)
1	TX0_UNDERFLOW	R/W	0h	Transmitter register underflow Channel 0 - (RW)
0	TX0_EMPTY	R/W	0h	Transmitter register empty or almost empty Channel 0 - (RW)

ADVANCE INFORMATION

13.7.1.8 IRQENABLE Register (Offset = 11Ch) [Reset = 00000000h]

IRQENABLE is shown in Table 13-19.

Return to the Summary Table.

This register allows to enable/disable the module internal sources of interrupt on an event-by-event basis.

Table 13-19. IRQENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reads return 0 - (RO)
17	EOW_ENABLE	R/W	0h	End of Word count Interrupt Enable - (RW)
16	WKE	R/W	0h	Wakeup event interrupt Enable in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CH0CONFSPIENSLV - (RW)
15	RESERVED	R	0h	Reads returns 0 - (RO)
14	RX3_FULL_ENABLE	R/W	0h	Receiver register Full Interrupt Enable Ch 3 - (RW)
13	TX3_UNDERFLOW_ENABLE	R/W	0h	Transmitter register Underflow Interrupt Enable Ch 3 - (RW)
12	TX3_EMPTY_ENABLE	R/W	0h	Transmitter register Empty Interrupt Enable Ch 3 - (RW)
11	RESERVED	R	0h	Reads return 0 - (RO)
10	RX2_FULL_ENABLE	R/W	0h	Receiver register Full Interrupt Enable Ch 2 - (RW)
9	TX2_UNDERFLOW_ENABLE	R/W	0h	Transmitter register Underflow Interrupt Enable Ch 2 - (RW)
8	TX2_EMPTY_ENABLE	R/W	0h	Transmitter register Empty Interrupt Enable Ch 2 - (RW)

Table 13-19. IRQENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RESERVED	R	0h	Reads return 0 - (RO)
6	RX1_FULL_ENABLE	R/W	0h	Receiver register Full Interrupt Enable Ch 1 - (RW)
5	TX1_UNDERFLOW_ENABLE	R/W	0h	Transmitter register Underflow Interrupt Enable Ch 1 - (RW)
4	TX1_EMPTY_ENABLE	R/W	0h	Transmitter register Empty Interrupt Enable Ch 1 - (RW)
3	RX0_OVERFLOW_ENABLE	R/W	0h	Receiver register Overflow Interrupt Enable Ch 0 - (RW)
2	RX0_FULL_ENABLE	R/W	0h	Receiver register Full Interrupt Enable Ch 0 - (RW)
1	TX0_UNDERFLOW_ENABLE	R/W	0h	Transmitter register Underflow Interrupt Enable Ch 0 - (RW)
0	TX0_EMPTY_ENABLE	R/W	0h	Transmitter register Empty Interrupt Enable Ch 0 - (RW)

13.7.1.9 WAKEUPENABLE Register (Offset = 120h) [Reset = 00000000h]

WAKEUPENABLE is shown in Table 13-20.

Return to the Summary Table.

The wakeup enable register allows to enable/disable the module internal sources of wakeup on event-by-event basis.

Table 13-20. WAKEUPENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reads returns 0 - (RO)
0	WKEN	R/W	0h	WakeUp functionality in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CH0CONFSPIENSLV - (RW)

13.7.1.10 SYST Register (Offset = 124h) [Reset = 00000000h]

SYST is shown in Table 13-21.

Return to the Summary Table.

This register is used to check the correctness of the system interconnect either internally to peripheral bus or externally to device IO pads when the module is configured in system test (SYSTEST) mode.

Table 13-21. SYST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reads returns 0 - (RO)
11	SSB	R/W	0h	Set status bit - (RW)
10	SPIENDIR	R/W	0h	Set the direction of the SPIEN [3:0] lines and SPICLK line - (RW)
9	SPIDATDIR1	R/W	0h	Set the direction of the SPIDAT[1] - (RW)
8	SPIDATDIR0	R/W	0h	Set the direction of the SPIDAT[0] - (RW)
7	WAKD	R/W	0h	SWAKEUP output [signal data value of internal signal to system] The signal is driven high or low according to the value written into this register bit - (RW)

ADVANCE INFORMATION

Table 13-21. SYST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SPICLK	R/W	0h	SPICLK line [signal data value] If MCSPI_SYSTSPIENDIR = 1 [input mode direction] this bit returns the value on the CLKSPI line [high or low] and a write into this bit has no effect If MCSPI_SYSTSPIENDIR = 0 [output mode direction] the CLKSPI line is driven high or low according to the value written into this register - (RW)
5	SPIDAT_1	R/W	0h	SPIDAT[1] line [signal data value] If MCSPI_SYSTSPIDATDIR1 = 0 [output mode direction] the SPIDAT[1] line is driven high or low according to the value written into this register If MCSPI_SYSTSPIDATDIR1 = 1 [input mode direction] this bit returns the value on the SPIDAT[1] line [high or low] and a write into this bit has no effect - (RW)
4	SPIDAT_0	R/W	0h	SPIDAT[0] line [signal data value] If MCSPI_SYSTSPIDATDIR0 = 0 [output mode direction] the SPIDAT[0] line is driven high or low according to the value written into this register If MCSPI_SYSTSPIDATDIR0 = 1 [input mode direction] this bit returns the value on the SPIDAT[0] line [high or low] and a write into this bit has no effect - (RW)
3	SPIEN_3	R/W	0h	SPIEN[3] line [signal data value] If MCSPI_SYSTSPIENDIR = 0 [output mode direction] the SPIENT[3] line is driven high or low according to the value written into this register If MCSPI_SYSTSPIENDIR = 1 [input mode direction] this bit returns the value on the SPIEN[3] line [high or low] and a write into this bit has no effect - (RW)
2	SPIEN_2	R/W	0h	SPIEN[2] line [signal data value] If MCSPI_SYSTSPIENDIR = 0 [output mode direction] the SPIENT[2] line is driven high or low according to the value written into this register If MCSPI_SYSTSPIENDIR = 1 [input mode direction] this bit returns the value on the SPIEN[2] line [high or low] and a write into this bit has no effect - (RW)
1	SPIEN_1	R/W	0h	SPIEN[1] line [signal data value] If MCSPI_SYSTSPIENDIR = 0 [output mode direction] the SPIENT[1] line is driven high or low according to the value written into this register If MCSPI_SYSTSPIENDIR = 1 [input mode direction] this bit returns the value on the SPIEN[1] line [high or low] and a write into this bit has no effect - (RW)
0	SPIEN_0	R/W	0h	SPIEN[0] line [signal data value] If MCSPI_SYSTSPIENDIR = 0 [output mode direction] the SPIENT[0] line is driven high or low according to the value written into this register If MCSPI_SYSTSPIENDIR = 1 [input mode direction] this bit returns the value on the SPIEN[0] line [high or low] and a write into this bit has no effect - (RW)

ADVANCE INFORMATION

13.7.1.11 MODULCTRL Register (Offset = 128h) [Reset = 00000004h]

MODULCTRL is shown in Table 13-22.

Return to the Summary Table.

This register is dedicated to the configuration of the serial port interface.

Table 13-22. MODULCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reads returns 0 - (RO)
8	FDAA	R/W	0h	FIFO DMA Address 256-bit aligned This register is used when a FIFO is managed by the module and DMA connected to the controller provides only 256 bit aligned address If this bit is set the enabled channel which uses the FIFO has its datas managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TX[i] and MCSPI_RX[i] registers - (RW)

Table 13-22. MODULCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MOA	R/W	0h	Multiple word ocp access: This register can only be used when a channel is enabled using a FIFO It allows the system to perform multiple SPI word access for a single 32-bit OCP word access This is possible for WL < 16 - (RW)
6-4	INITDLY	R/W	0h	Initial spi delay for first transfer: This register is an option only available in SINGLE master mode The controller waits for a delay to transmit the first spi word after channel enabled and corresponding TX register filled This Delay is based on SPI output frequency clock No clock output provided to the boundary and chip select is not active in 4 pin mode within this period - (RW)
3	SYSTEM_TEST	R/W	0h	Enables the system test mode - (RW)
2	MS	R/W	1h	Master/ Slave - (RW)
1	PIN34	R/W	0h	Pin mode selection: This register is used to configure the SPI pin mode in master or slave mode If asserted the controller only use SIMOSOMI and SPICLK clock pin for spi transfers - (RW)
0	SINGLE	R/W	0h	Single channel / Multi Channel [master mode only] - (RW)

13.7.1.12 CH0CONF Register (Offset = 12Ch) [Reset = 00060000h]

CH0CONF is shown in Table 13-23.

Return to the Summary Table.

This register is dedicated to the configuration of the channel 0

Table 13-23. CH0CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	read returns 0 - (RO)
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values - (RW)
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set - (RW)
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set - (RW)
26-25	TCS0	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock - (RW)
24	SBPOL	R/W	0h	Start bit polarity - (RW)
23	SBE	R/W	0h	Start bit enable for SPI transfer - (RW)
22-21	SPIENSLV	R/W	0h	Channel 0 only and slave mode only: SPI slave select signal detection Reserved bits for other cases - (RW)
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words [single channel master mode only] - (RW)
19	TURBO	R/W	0h	Turbo mode - (RW)
18	IS	R/W	1h	Input Select - (RW)
17	DPE1	R/W	1h	Transmission Enable for data line 1 [SPIDATAGZEN[1]] - (RW)
16	DPE0	R/W	0h	Transmission Enable for data line 0 [SPIDATAGZEN[0]] - (RW)
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel The DMA Read request line is deasserted on read completion of the receive register of the channel - (RW)

Table 13-23. CH0CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	DMAW	R/W	0h	DMA Write request The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty The DMA Write request line is deasserted on load completion of the transmitter register of the channel - (RW)
13-12	TRM	R/W	0h	Transmit/Receive modes - (RW)
11-7	WL	R/W	0h	SPI word length - (RW)
6	EPOL	R/W	0h	SPIEN polarity - (RW)
5-2	CLKD	R/W	0h	Frequency divider for SPICLK [only when the module is a Master SPI device] A programmable clock divider divides the SPI reference clock [CLKSPIREF] with a 4-bit value and results in a new clock SPICLK available to shift-in and shift-out data By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] registerThe value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0 - (RW)
1	POL	R/W	0h	SPICLK polarity - (RW)
0	PHA	R/W	0h	SPICLK phase - (RW)

13.7.1.13 CH0STAT Register (Offset = 130h) [Reset = 00000000h]

CH0STAT is shown in Table 13-24.

Return to the Summary Table.

This register provides status information about transmitter and receiver registers of channel 0

Table 13-24. CH0STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0 - (RO)
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status - (RO)
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status - (RO)
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status - (RO)
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status - (RO)
2	EOT	R	0h	Channel "i" End of transfer Status The definitions of beginning and end of transfer vary with master versus slave and the transfer format [Transmit/Receive modes Turbo mode] See dedicated chapters for details - (RO)
1	TXS	R	0h	Channel "i" Transmitter Register Status - (RO)
0	RXS	R	0h	Channel "i" Receiver Register Status - (RO)

13.7.1.14 CH0CTRL Register (Offset = 134h) [Reset = 00000000h]

CH0CTRL is shown in Table 13-25.

Return to the Summary Table.

This register is dedicated to enable the channel 0

Table 13-25. CH0CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Read returns 0 - (RO)

Table 13-25. CH0CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle [MCSPI_CHCONF[CLKG] set to 1] Then the max value reached is 4096 clock divider ratio - (RW)
7-1	RESERVED	R	0h	Read returns 0 - (RO)
0	EN	R/W	0h	Channel Enable - (RW)

13.7.1.15 TX0 Register (Offset = 138h) [Reset = 00000000h]

TX0 is shown in Table 13-26.

Return to the Summary Table.

This register contains a single SPI word to transmit on the serial link what ever SPI word length is.

Table 13-26. TX0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 0 Data to transmit - (RW)

13.7.1.16 RX0 Register (Offset = 13Ch) [Reset = 00000000h]

RX0 is shown in Table 13-27.

Return to the Summary Table.

This register contains a single SPI word received through the serial link what ever SPI word length is.

Table 13-27. RX0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 0 Received Data - (RO)

13.7.1.17 CH1CONF Register (Offset = 140h) [Reset = 00060000h]

CH1CONF is shown in Table 13-28.

Return to the Summary Table.

This register is dedicated to the configuration of the channel.

Table 13-28. CH1CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	read returns 0 - (RO)
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values - (RW)
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set - (RW)
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set - (RW)
26-25	TCS1	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock - (RW)
24	SBPOL	R/W	0h	Start bit polarity - (RW)

Table 13-28. CH1CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	SBE	R/W	0h	Start bit enable for SPI transfer - (RW)
22-21	RESERVED	R	0h	read returns 0 - (RO)
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words [single channel master mode only] - (RW)
19	TURBO	R/W	0h	Turbo mode - (RW)
18	IS	R/W	1h	Input Select - (RW)
17	DPE1	R/W	1h	Transmission Enable for data line 1 [SPIDATAGZEN[1]] - (RW)
16	DPE0	R/W	0h	Transmission Enable for data line 0 [SPIDATAGZEN[0]] - (RW)
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel The DMA Read request line is deasserted on read completion of the receive register of the channel - (RW)
14	DMAW	R/W	0h	DMA Write request The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty The DMA Write request line is deasserted on load completion of the transmitter register of the channel - (RW)
13-12	TRM	R/W	0h	Transmit/Receive modes - (RW)
11-7	WL	R/W	0h	SPI word length - (RW)
6	EPOL	R/W	0h	SPIEN polarity - (RW)
5-2	CLKD	R/W	0h	Frequency divider for SPICLK [only when the module is a Master SPI device] A programmable clock divider divides the SPI reference clock [CLKSPIREF] with a 4-bit value and results in a new clock SPICLK available to shift-in and shift-out data By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] registerThe value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0 - (RW)
1	POL	R/W	0h	SPICLK polarity - (RW)
0	PHA	R/W	0h	SPICLK phase - (RW)

13.7.1.18 CH1STAT Register (Offset = 144h) [Reset = 00000000h]

CH1STAT is shown in Table 13-29.

Return to the Summary Table.

This register provides status information about transmitter and receiver registers of channel 1

Table 13-29. CH1STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0 - (RO)
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status - (RO)
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status - (RO)
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status - (RO)
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status - (RO)
2	EOT	R	0h	Channel "i" End of transfer Status The definitions of beginning and end of transfer vary with master versus slave and the transfer format [Transmit/Receive modes Turbo mode] See dedicated chapters for details - (RO)
1	TXS	R	0h	Channel "i" Transmitter Register Status - (RO)

Table 13-29. CH1STAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RXS	R	0h	Channel "i" Receiver Register Status - (RO)

13.7.1.19 CH1CTRL Register (Offset = 148h) [Reset = 00000000h]

CH1CTRL is shown in Table 13-30.

Return to the Summary Table.

This register is dedicated to enable the channel 1

Table 13-30. CH1CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Read returns 0 - (RO)
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle [MCSPI_CHCONF[CLKG] set to 1] Then the max value reached is 4096 clock divider ratio - (RW)
7-1	RESERVED	R	0h	Read returns 0 - (RO)
0	EN	R/W	0h	Channel Enable - (RW)

13.7.1.20 TX1 Register (Offset = 14Ch) [Reset = 00000000h]

TX1 is shown in Table 13-31.

Return to the Summary Table.

This register contains a single SPI word to transmit on the serial link what ever SPI word length is.

Table 13-31. TX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 1 Data to transmit - (RW)

13.7.1.21 RX1 Register (Offset = 150h) [Reset = 00000000h]

RX1 is shown in Table 13-32.

Return to the Summary Table.

This register contains a single SPI word received through the serial link what ever SPI word length is.

Table 13-32. RX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 1 Received Data - (RO)

13.7.1.22 CH2CONF Register (Offset = 154h) [Reset = 00060000h]

CH2CONF is shown in Table 13-33.

Return to the Summary Table.

This register is dedicated to the configuration of the channel 2

Table 13-33. CH2CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	read returns 0 - (RO)
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values - (RW)
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set - (RW)
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set - (RW)
26-25	TCS2	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock - (RW)
24	SBPOL	R/W	0h	Start bit polarity - (RW)
23	SBE	R/W	0h	Start bit enable for SPI transfer - (RW)
22-21	RESERVED	R	0h	read returns 0 - (RO)
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words [single channel master mode only] - (RW)
19	TURBO	R/W	0h	Turbo mode - (RW)
18	IS	R/W	1h	Input Select - (RW)
17	DPE1	R/W	1h	Transmission Enable for data line 1 [SPIDATAGZEN[1]] - (RW)
16	DPE0	R/W	0h	Transmission Enable for data line 0 [SPIDATAGZEN[0]] - (RW)
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel The DMA Read request line is deasserted on read completion of the receive register of the channel - (RW)
14	DMAW	R/W	0h	DMA Write request The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty The DMA Write request line is deasserted on load completion of the transmitter register of the channel - (RW)
13-12	TRM	R/W	0h	Transmit/Receive modes - (RW)
11-7	WL	R/W	0h	SPI word length - (RW)
6	EPOL	R/W	0h	SPIEN polarity - (RW)
5-2	CLKD	R/W	0h	Frequency divider for SPICLK [only when the module is a Master SPI device] A programmable clock divider divides the SPI reference clock [CLKSPIREF] with a 4-bit value and results in a new clock SPICLK available to shift-in and shift-out data By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] registerThe value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0 - (RW)
1	POL	R/W	0h	SPICLK polarity - (RW)
0	PHA	R/W	0h	SPICLK phase - (RW)

13.7.1.23 CH2STAT Register (Offset = 158h) [Reset = 00000000h]

CH2STAT is shown in Table 13-34.

Return to the Summary Table.

This register provides status information about transmitter and receiver registers of channel 2

Table 13-34. CH2STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0 - (RO)
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status - (RO)
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status - (RO)
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status - (RO)
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status - (RO)
2	EOT	R	0h	Channel "i" End of transfer Status The definitions of beginning and end of transfer vary with master versus slave and the transfer format [Transmit/Receive modes Turbo mode] See dedicated chapters for details - (RO)
1	TXS	R	0h	Channel "i" Transmitter Register Status - (RO)
0	RXS	R	0h	Channel "i" Receiver Register Status - (RO)

13.7.1.24 CH2CTRL Register (Offset = 15Ch) [Reset = 00000000h]

CH2CTRL is shown in Table 13-35.

Return to the Summary Table.

This register is dedicated to enable the channel 2

Table 13-35. CH2CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Read returns 0 - (RO)
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle [MCSPI_CHCONF[CLKG] set to 1] Then the max value reached is 4096 clock divider ratio - (RW)
7-1	RESERVED	R	0h	Read returns 0 - (RO)
0	EN	R/W	0h	Channel Enable - (RW)

13.7.1.25 TX2 Register (Offset = 160h) [Reset = 00000000h]

TX2 is shown in Table 13-36.

Return to the Summary Table.

This register contains a single SPI word to transmit on the serial link what ever SPI word length is.

Table 13-36. TX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 2 Data to transmit - (RW)

13.7.1.26 RX2 Register (Offset = 164h) [Reset = 00000000h]

RX2 is shown in Table 13-37.

Return to the Summary Table.

This register contains a single SPI word received through the serial link what ever SPI word length is.

Table 13-37. RX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 2 Received Data - (RO)

13.7.1.27 CH3CONF Register (Offset = 168h) [Reset = 00060000h]

CH3CONF is shown in Table 13-38.

Return to the Summary Table.

This register is dedicated to the configuration of the channel 3

Table 13-38. CH3CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	read returns 0 - (RO)
29	CLKG	R/W	0h	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values - (RW)
28	FFER	R/W	0h	FIFO enabled for receive:Only one channel can have this bit field set - (RW)
27	FFEW	R/W	0h	FIFO enabled for Transmit:Only one channel can have this bit field set - (RW)
26-25	TCS3	R/W	0h	Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock - (RW)
24	SBPOL	R/W	0h	Start bit polarity - (RW)
23	SBE	R/W	0h	Start bit enable for SPI transfer - (RW)
22-21	RESERVED	R	0h	read returns 0 - (RO)
20	FORCE	R/W	0h	Manual SPIEN assertion to keep SPIEN active between SPI words [single channel master mode only] - (RW)
19	TURBO	R/W	0h	Turbo mode - (RW)
18	IS	R/W	1h	Input Select - (RW)
17	DPE1	R/W	1h	Transmission Enable for data line 1 [SPIDATAGZEN[1]] - (RW)
16	DPE0	R/W	0h	Transmission Enable for data line 0 [SPIDATAGZEN[0]] - (RW)
15	DMAR	R/W	0h	DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel The DMA Read request line is deasserted on read completion of the receive register of the channel - (RW)
14	DMAW	R/W	0h	DMA Write request The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty The DMA Write request line is deasserted on load completion of the transmitter register of the channel - (RW)
13-12	TRM	R/W	0h	Transmit/Receive modes - (RW)
11-7	WL	R/W	0h	SPI word length - (RW)
6	EPOL	R/W	0h	SPIEN polarity - (RW)

Table 13-38. CH3CONF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-2	CLKD	R/W	0h	Frequency divider for SPICLK [only when the module is a Master SPI device] A programmable clock divider divides the SPI reference clock [CLKSPIREF] with a 4-bit value and results in a new clock SPICLK available to shift-in and shift-out data By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] registerThe value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0 - (RW)
1	POL	R/W	0h	SPICLK polarity - (RW)
0	PHA	R/W	0h	SPICLK phase - (RW)

13.7.1.28 CH3STAT Register (Offset = 16Ch) [Reset = 00000000h]

CH3STAT is shown in Table 13-39.

Return to the Summary Table.

This register provides status information about transmitter and receiver registers of channel 3

Table 13-39. CH3STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Read returns 0 - (RO)
6	RXFFF	R	0h	Channel "i" FIFO Receive Buffer Full Status - (RO)
5	RXFFE	R	0h	Channel "i" FIFO Receive Buffer Empty Status - (RO)
4	TXFFF	R	0h	Channel "i" FIFO Transmit Buffer Full Status - (RO)
3	TXFFE	R	0h	Channel "i" FIFO Transmit Buffer Empty Status - (RO)
2	EOT	R	0h	Channel "i" End of transfer Status The definitions of beginning and end of transfer vary with master versus slave and the transfer format [Transmit/Receive modes Turbo mode] See dedicated chapters for details - (RO)
1	TXS	R	0h	Channel "i" Transmitter Register Status - (RO)
0	RXS	R	0h	Channel "i" Receiver Register Status - (RO)

13.7.1.29 CH3CTRL Register (Offset = 170h) [Reset = 00000000h]

CH3CTRL is shown in Table 13-40.

Return to the Summary Table.

This register is dedicated to enable the channel 3

Table 13-40. CH3CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Read returns 0 - (RO)
15-8	EXTCLK	R/W	0h	Clock ratio extension: This register is used to concatenate with MCSPI_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle [MCSPI_CHCONF[CLKG] set to 1] Then the max value reached is 4096 clock divider ratio - (RW)
7-1	RESERVED	R	0h	Read returns 0 - (RO)
0	EN	R/W	0h	Channel Enable - (RW)

13.7.1.30 TX3 Register (Offset = 174h) [Reset = 00000000h]

TX3 is shown in Table 13-41.

Return to the Summary Table.

This register contains a single SPI word to transmit on the serial link what ever SPI word length is.

Table 13-41. TX3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TDATA	R/W	0h	Channel 3 Data to transmit - (RW)

13.7.1.31 RX3 Register (Offset = 178h) [Reset = 00000000h]

RX3 is shown in Table 13-42.

Return to the Summary Table.

This register contains a single SPI word received through the serial link what ever SPI word length is.

Table 13-42. RX3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RDATA	R	0h	Channel 3 Received Data - (RO)

13.7.1.32 XFERLEVEL Register (Offset = 17Ch) [Reset = 00000000h]

XFERLEVEL is shown in Table 13-43.

Return to the Summary Table.

This register provides transfer levels needed while using FIFO buffer during transfer.

Table 13-43. XFERLEVEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	WCNT	R/W	0h	Spi word counterThis register holds the programmable value of number of SPI word to be transferred on channel which is using the FIFO bufferWhen transfer had started a read back in this register returns the current SPI word transfer index - (RW)
15-8	AFL	R/W	0h	Buffer Almost Full This register holds the programmable almost full level value used to determine almost full buffer condition If the user wants an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least n bytes then the buffer MCSPI_MODULCTRLAFL must be set with n-1The size of this register is defined by the generic parameter FFNBYTE - (RW)
7-0	AEL	R/W	0h	Buffer Almost EmptyThis register holds the programmable almost empty level value used to determine almost empty buffer condition If the user wants an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive n bytes then the buffer MCSPI_MODULCTRLAEL must be set with n-1 - (RW)

13.7.1.33 DAFTX Register (Offset = 180h) [Reset = 00000000h]

DAFTX is shown in Table 13-44.

Return to the Summary Table.

This register contains the SPI words to transmit on the serial link when FIFO used and DMA address is aligned on 256 bit.This register is an image of one of MCSPI_TX(i) register corresponding to the channel which have its FIFO enabled.

Table 13-44. DAFTX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DAFTDATA	R/W	0h	FIFO Data to transmit with DMA 256 bit aligned address This Register is only is used when MCSPI_MODULCTRLFDAA is set to "1" and only one of the MCSPI_CH[i]CONF[FFEW] of enabled channels is set If these conditions are not respected any access to this register return a null value - (RW)

13.7.1.34 DAFRX Register (Offset = 1A0h) [Reset = 00000000h]

DAFRX is shown in Table 13-45.

Return to the Summary Table.

This register contains the SPI words to received on the serial link when FIFO used and DMA address is aligned on 256 bit.This register is an image of one of MCSPI_RX(i) register corresponding to the channel which have its FIFO enabled.

Table 13-45. DAFRX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DAFRDATA	R	0h	FIFO Data to transmit with DMA 256 bit aligned address This Register is only is used when MCSPI_MODULCTRLFDAA is set to "1" and only one of the MCSPI_CH[i]CONF[FFEW] of enabled channels is set If these conditions are not respected any access to this register return a null value - (RO)

14 Real-Time Interrupt (RTI) and Watchdog Module

This chapter 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).

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 datasheet to determine the applicability of these features to your device being used.

14.1 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 time bases 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.

14.1.1 Features

The RTI module has the following features:

- Two independent 64 bit counter blocks
- Windowed Watchdog Timer (WWDT) Feature
- 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

The RTI does not support the following features:

- External clock supervising circuit to switch to internal prescale counter 0, if external clock source fails to increment in a predefined window.
- Capture events to capture timestamps through recording of timer status.
- Two time-stamp (capture) functions for system or peripheral interrupts, one for each counter block.
- Analog Watchdog via external RC Network to prevent for runaway code.

14.1.2 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.

14.2 Module Operation

Figure 14-1 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. The two counter blocks provide the same basic functionality.

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.

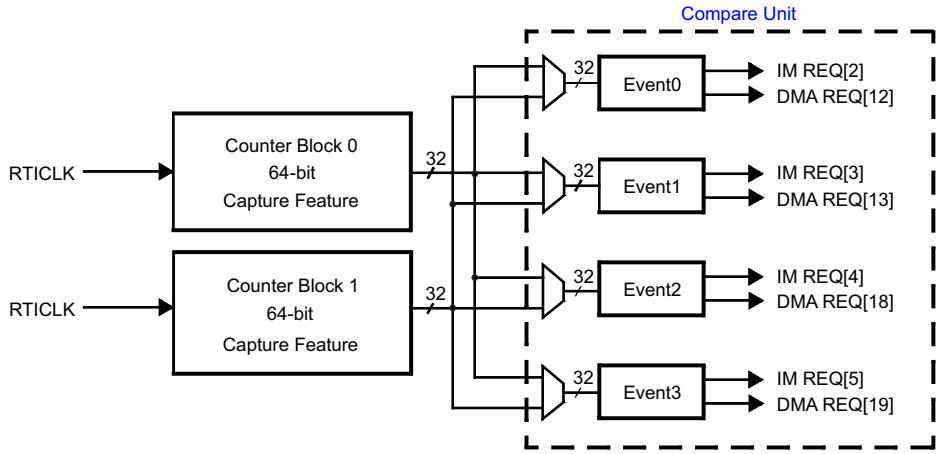


Figure 14-1. RTI Block Diagram

Note

During deep sleep entry, there is a need to disable the WDT warm reset propagation.

The WDT does **not** provide coverage in deep sleep mode. Thus, proper care must be taken by the application to ensure the WDT is not enabled when entering this mode. TOP_PRCM:RST_WDT_RESET_EN[0] must be set to 0.

14.2.1 Counter Operation

Each counter block consists of the following (see Figure 14-2):

- 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 interrupt manager (NVIC/IM). The overflow interrupt is not intended to generate the timebase for the operating system. See Section 14.2.2 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:

$$f_{RTIFRCx} = \begin{cases} \dfrac{f_{RTICLK}}{RTICPUCx + 1} & \text{when RTICPUCx} \neq 0 \\ \dfrac{f_{RTICLK}}{2^{32}+1} & \text{when RTICPUCx} = 0 \end{cases} \quad (2)$$

ADVANCE INFORMATION

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.

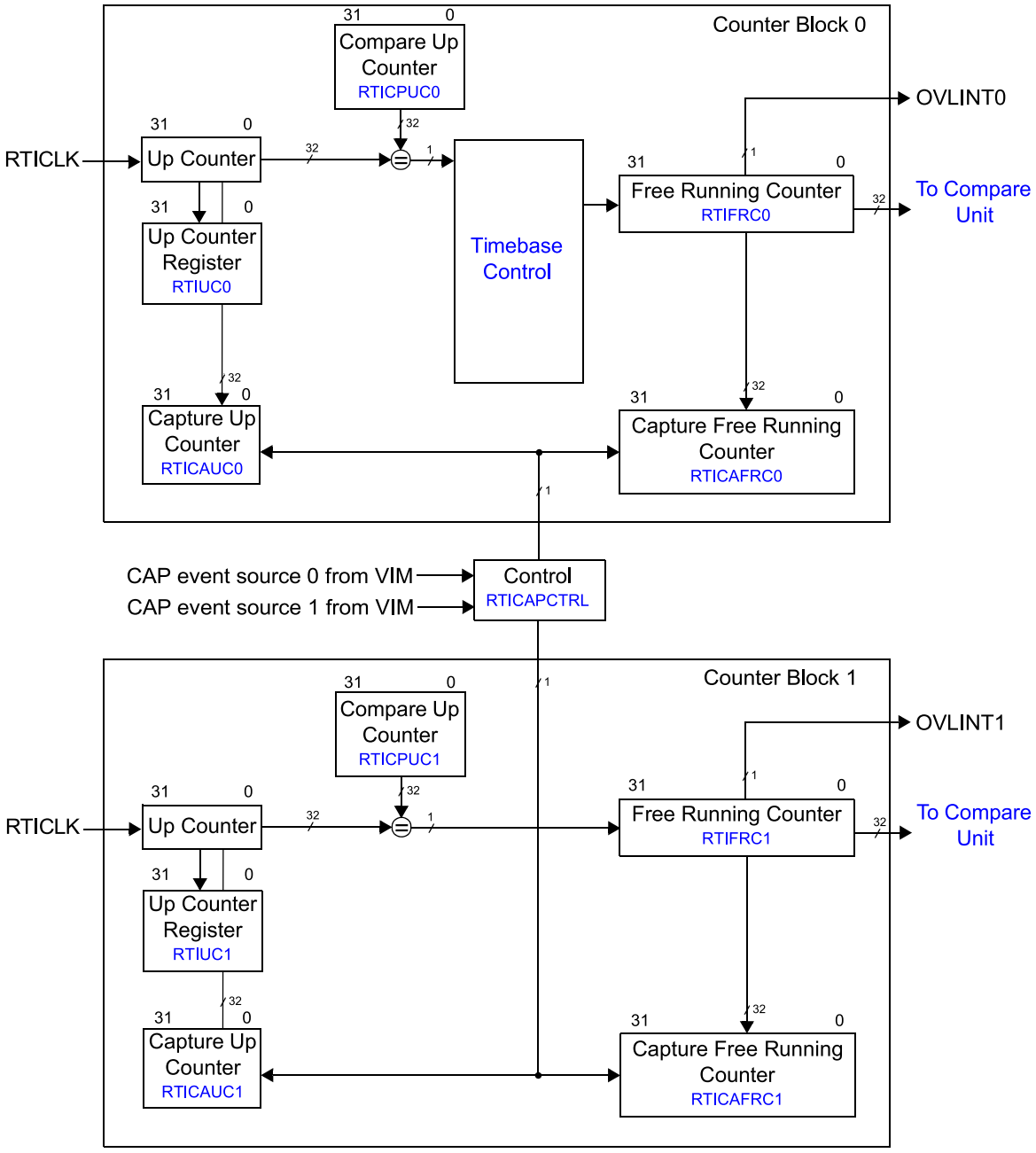


Figure 14-2. Counter Block Diagram

14.2.1.1 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.

14.2.1.2 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 Interrupt Manager (NVIC/IM) 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 NVIC/IM.

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 14.2.1.1).

14.2.2 Interrupt/DMA Requests

There are four compare registers (RTICOMPy) to generate interrupt requests to the NVIC/IM 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,

$$t_{COMPx} = t_{RTICLK} \times (RTICPUCy + 1) \times 2^{32} \quad (3)$$

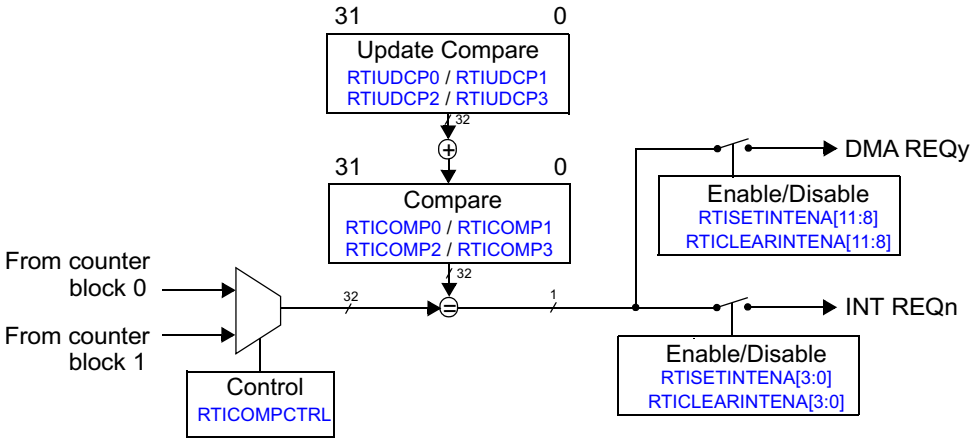


Figure 14-3. Compare Unit Block Diagram (shows only 1 of 4 blocks for simplification)

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.

14.2.3 RTI Clocking

The counter blocks are clocked with RTICLK.

A clock supervision for the NTUx clocking scheme is implemented to avoid missing operating system ticks.

14.2.4 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 14-4 illustrates the DWD.

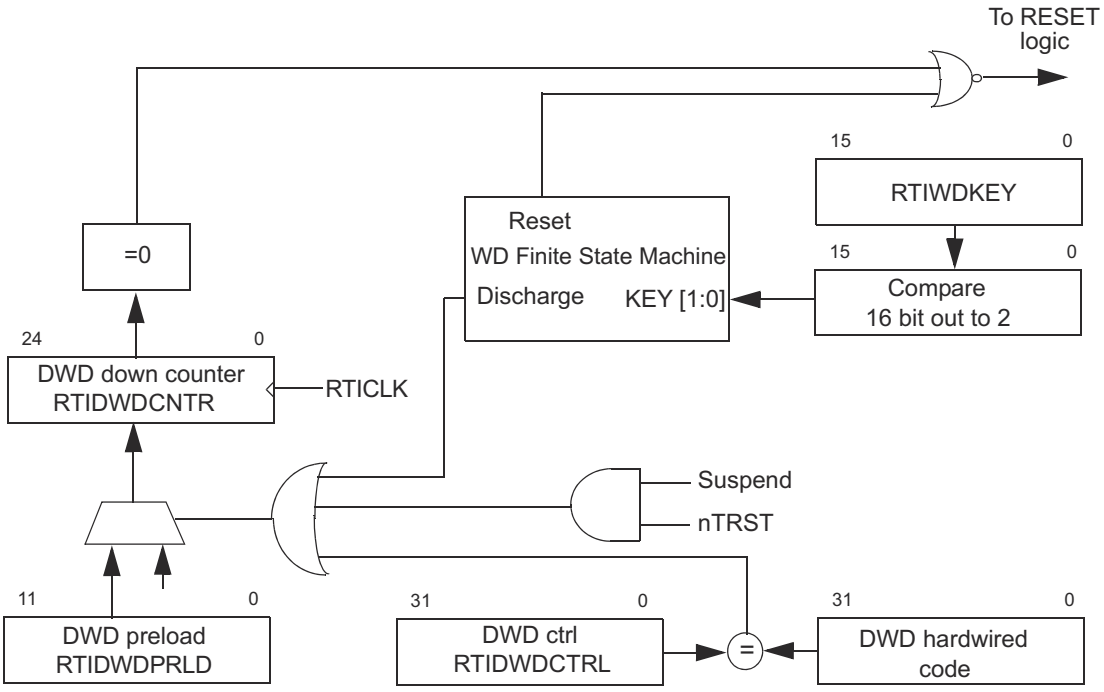


Figure 14-4. Digital Watchdog

ADVANCE INFORMATION

14.2.4.1 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.

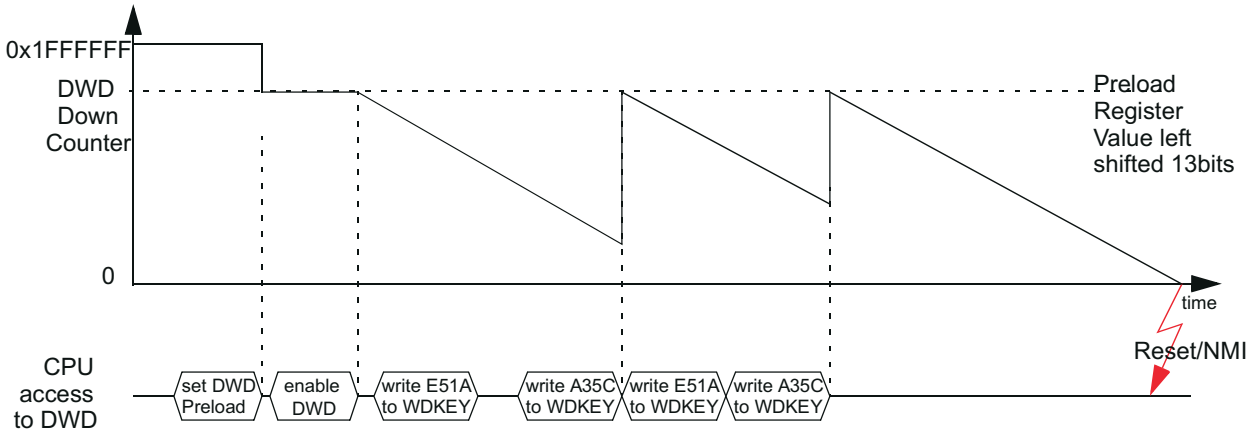


Figure 14-5. DWD Operation

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.

14.2.4.2 Digital Windowed Watchdog (DWWD)

In addition to the time-out boundary configurable via the digital watchdog discussed in Section 14.2.4.1, 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 14-6. 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 14-7 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.

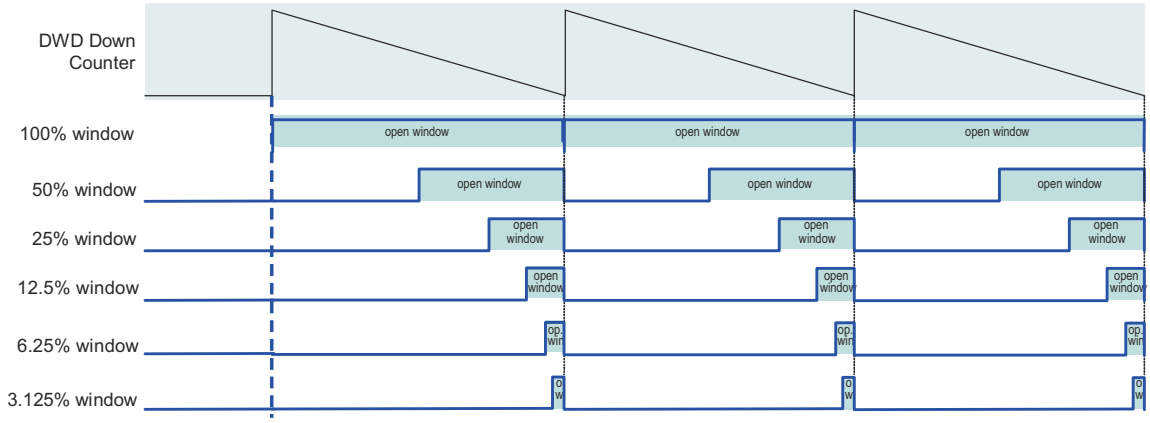


Figure 14-6. Digital Windowed Watchdog Timing Example

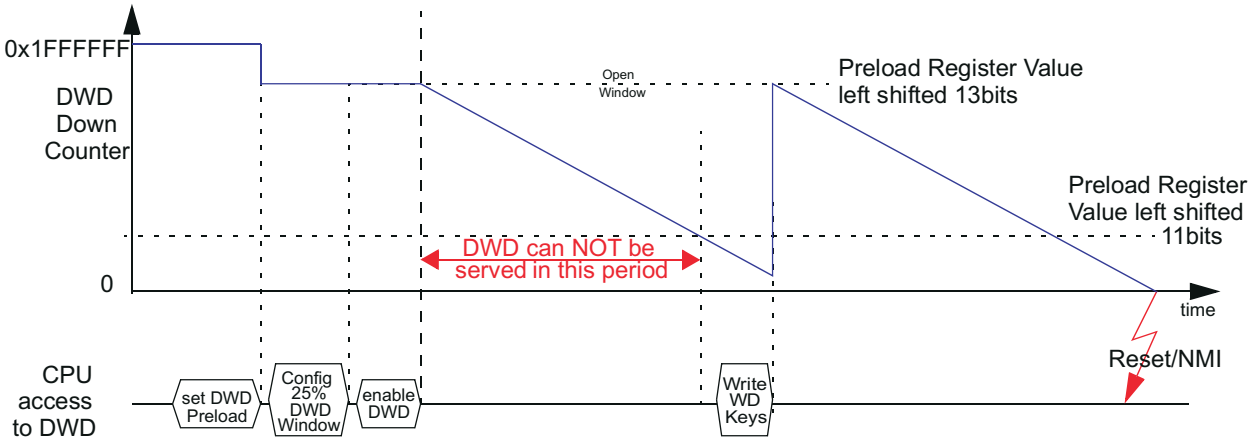


Figure 14-7. Digital Windowed Watchdog Operation Example (25% Window)

14.2.5 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. However, if the external timebase (NTU) is used and the system is in halting debug mode, the timebase control circuit will switch to internal timebase once it detects the missing NTU signal of the suspended communication controller. This will be signaled with an TBINT interrupt so that software can resynchronize after the device exits halting debug mode.

14.3 APPSS RTI Integration Details

Event Capture

The APPSS RTI Capture Event mapping can be selected from the APPSS Interrupt Map.

- Interrupt lines [63:0] can be selected and are the legal values that can be configured.
- APPSS_CTRL:: APPSS_IRQ_REQ_SEL::CAPEVTSEL_RTI_SRC0 is used to select RTI and WDT Capture Event 0
- APPSS_CTRL:: APPSS_IRQ_REQ_SEL::CAPEVTSEL_RTI_SRC1 is used to select RTI and WDT Capture Event 1

Emulation Suspend

Suspends for RTI/WDT in APPSS are masked with following control signal.

- APPSS_CTRL:: APPSS_DBG_ACK_CTL::DBG_ACK_CTL1_RTI is used to mask the suspend signal for APPSS RTI
- APPSS_CTRL:: APPSS_DBG_ACK_CTL::DBG_ACK_CTL1_WDT is used to mask the suspend signal for APPSS WDT

Interrupts

Each RTI generates below interrupts to CM4

- APPSS_RTI/WDT/_INT0/1/2/3
- APPSS_RTI/WDT/_OVERFLOW_INT0 /1 (Overflow)
- APPSS_RTI/WDT/_TB_INT (Time Base Interrupt)

Errors

- No Errors are generated by any RTI

DMA REQs

Each RTI generated below dma requests to APPSS_TPCC_A in the APPSS. The RTI/WDT dma req is not used for HWA_TPCC.

- APPSS_RTI/WDT/_DMA_REQ0/1/2/3

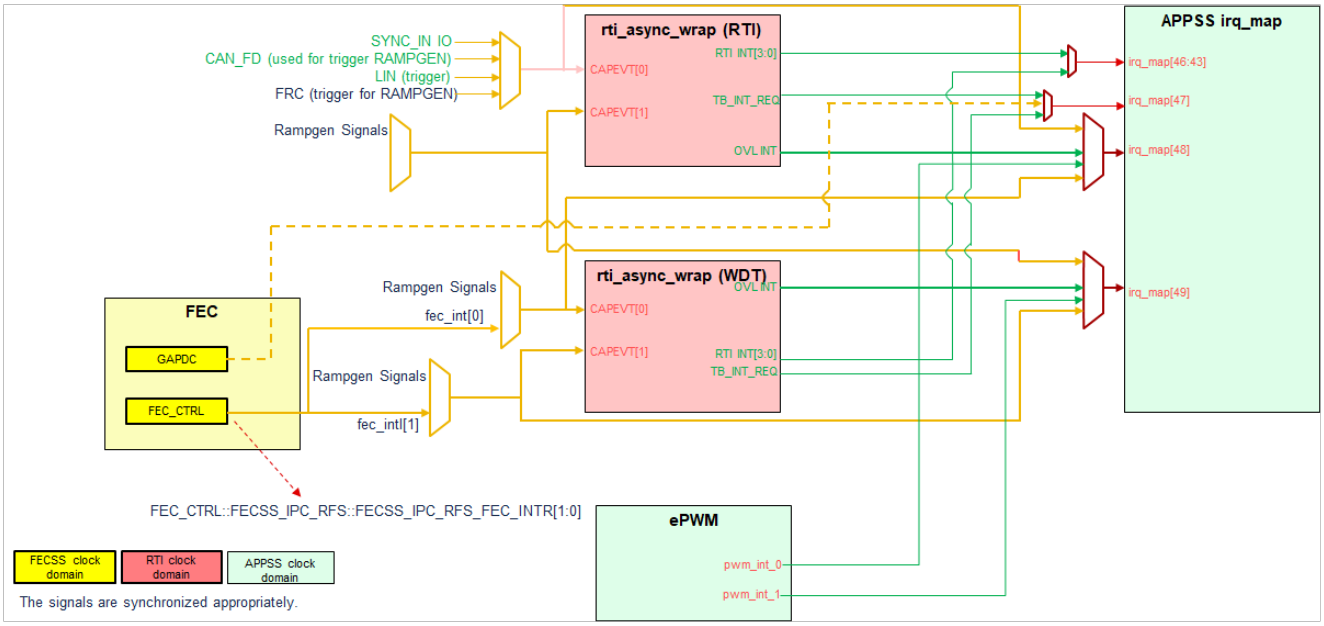


Figure 14-8. APPSS RTI

14.4 APP_RTI Registers

Table 14-1 lists the memory-mapped registers for the APP_RTI registers. All register offset addresses not listed in Table 14-1 should be considered as reserved locations and the register contents should not be modified.

Table 14-1. APP_RTI Registers

Offset	Acronym	Register Name	Section
0h	RTIGCTRL	RTIGCTRL	Go
4h	RTITBCTRL	RTITBCTRL	Go
8h	RTICAPCTRL	RTICAPCTRL	Go
Ch	RTICOMPCTRL	RTICOMPCTRL	Go
10h	RTIFRC0	RTIFRC0	Go
14h	RTIUC0	RTIUC0	Go
18h	RTICPUC0	RTICPUC0	Go
20h	RTICAFRC0	RTICAFRC0	Go
24h	RTICAUC0	RTICAUC0	Go
30h	RTIFRC1	RTIFRC1	Go
34h	RTIUC1	RTIUC1	Go
38h	RTICPUC1	RTICPUC1	Go
40h	RTICAFRC1	RTICAFRC1	Go
44h	RTICAUC1	RTICAUC1	Go
50h	RTICOMP0	RTICOMP0	Go
54h	RTIUDCP0	RTIUDCP0	Go
58h	RTICOMP1	RTICOMP1	Go
5Ch	RTIUDCP1	RTIUDCP1	Go
60h	RTICOMP2	RTICOMP2	Go
64h	RTIUDCP2	RTIUDCP2	Go
68h	RTICOMP3	RTICOMP3	Go
6Ch	RTIUDCP3	RTIUDCP3	Go
70h	RTITBLCOMP	RTITBLCOMP	Go
74h	RTITBHCOMP	RTITBHCOMP	Go
80h	RTISETINT	RTISETINT	Go
84h	RTICLEARINT	RTICLEARINT	Go
88h	RTIINTFLAG	RTIINTFLAG	Go
90h	RTIDWDCTRL	RTIDWDCTRL	Go
94h	RTIDWDPRLD	RTIDWDPRLD	Go
98h	RTIWDSTATUS	RTIWDSTATUS	Go
9Ch	RTIWDKEY	RTIWDKEY	Go
A0h	RTIDWDCNTR	RTIDWDCNTR	Go
A4h	RTIWWDRXNCTRL	RTIWWDRXNCTRL	Go
A8h	RTIWWDSIZECTRL	RTIWWDSIZECTRL	Go
ACh	RTIINTCLRENABLE	RTIINTCLRENABLE	Go
B0h	RTICOMP0CLR	RTICOMP0CLR	Go
B4h	RTICOMP1CLR	RTICOMP1CLR	Go
B8h	RTICOMP2CLR	RTICOMP2CLR	Go
BCh	RTICOMP3CLR	RTICOMP3CLR	Go

Complex bit access types are encoded to fit into small table cells. Table 14-2 shows the codes that are used for access types in this section.

Table 14-2. APP_RTI 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

14.4.1 RTIGCTRL Register (Offset = 0h) [Reset = 00000000h]

RTIGCTRL is shown in Table 14-3.

Return to the Summary Table.

Global Control Register starts / stops the counters

Table 14-3. RTIGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
19-16	NTUSEL	R/W	0h	NTUSEL: Select NTU signal. These bits determine which NTU input signal is used as external timebase. There are up to four inputs supported with four valid selection combinations. Any invalid selection value written to the NTUSEL bit-field will result in a TIED LOW being used as the NTU signal. The NTU signal will also be TIED LOW in case of a single-bit flip as it will result in an invalid combination of NTUSEL. User and privilege mode (read): 0000 = NTU0 0101 = NTU1 1010 = NTU2 1111 = NTU3 other = tied to ‘0’ Privilege mode (write): 0000 = NTU0 0101 = NTU1 1010 = NTU2 1111 = NTU3 other = tied to ‘0’
15	COS	R/W	0h	COS: Continue On Suspend. This bit determines if both counters are stopped when the device goes into debug mode or if they continue counting. User and privilege mode (read): 0 = counters are stopped while in debug mode 1 = counters are running while in debug mode Privilege mode (write): 0 = stop counters in debug mode 1 = continue counting in debug mode
14-2	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
1	CNT1EN	R/W	0h	CNT1EN: Counter 1 Enable. The CNT1EN bit starts and stops the operation of counter block 1 (UC1 and FRC1). User and privilege mode (read): 0 = counters are stopped 1 = counters are running Privilege mode (write): 0 = stop counters 1 = start counters Gives the absolute 32 bit destination address (physical).

ADVANCE INFORMATION

Table 14-3. RTIGCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CNT0EN	R/W	0h	CNT0EN: Counter 0 Enable.The CNT0EN bit starts and stops the operation of counter block 0 (UC0 and FRC0). User and privilege mode (read): 0 = counters are stopped 1 = counters are running Privilege mode (write): 0 = stop counters 1 = start counters Gives the absolute 32 bits source address (physical).

14.4.2 RTITBCTRL Register (Offset = 4h) [Reset = 00000000h]

RTITBCTRL is shown in Table 14-4.

Return to the Summary Table.

Timebase Control selection which source triggers free running counter 0

Table 14-4. RTITBCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	Reserved
1	INC	R/W	0h	INC: Increment Free Running Counter 0. This bit determines wether the Free Running Counter 0 is automatically incremented if a failing clock on the NTUx signal is detected. User and privilege mode (read): 0 = FRC0 will not be incremented 1 = FRC0 will be incremented Privilege mode (write): 0 = Do not increment FRC0 on failing external clock 1 = Increment FRC0 on failing external clock
0	TBEXT	R/W	0h	TBEXT: Timebase External. The Timebase External bit selects whether the Free Running Counter 0 is clocked by the internal Up Counter 0 or from the external signal NTUx. Since setting the TBEXT bit to 1 resets Up Counter 0, Free Running Counter 0 will not be incremented in this occurence. The only source which is able to increment Free Running Counter 0 is NTUx. When the Timebase Supervisor circuit detects a missing clockedge, then the TBEXT bit is reset. The selection if the external signal should be used, can only be done by software. User and privilege mode (read): 0 = UC0 clocks FRC0 1 = NTUx clocks FRC0 Privilege mode (write): 0 = MUX is switched to internal UC0 clocking scheme 1 = MUX is switched to external NTUx clocking scheme

14.4.3 RTICAPCTRL Register (Offset = 8h) [Reset = 00000000h]

RTICAPCTRL is shown in Table 14-5.

Return to the Summary Table.

Capture Control controls the capture source for the counters

Table 14-5. RTICAPCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect

Table 14-5. RTICAPCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CAPCNTR1	R/W	0h	CAPCNTR 1: Capture Counter 1. This bit determines, which external interrupt source triggers a capture event of both UC1 and FRC1. User and privilege mode (read): 0 = capture event is triggered by Capture Event Source 0 1 = capture event is triggered by Capture Event Source 1 Privilege mode (write): 0 = enable capture event triggered by Capture Event Source 0 1 = enable capture event triggered by Capture Event Source 1
0	CAPCNTR0	R/W	0h	CAPCNTR 0: Capture Counter 0. This bit determines, which external interrupt source triggers a capture event of both UC0 and FRC0. User and privilege mode (read): 0 = capture event is triggered by Capture Event Source 0 1 = capture event is triggered by Capture Event Source 1 Privilege mode (write): 0 = enable capture event triggered by Capture Event Source 0 1 = enable capture event triggered by Capture Event Source 1 11 indexed 10 reserved 01 post-increment 00 constant

14.4.4 RTICOMPCTRL Register (Offset = Ch) [Reset = 00000000h]

RTICOMPCTRL is shown in Table 14-6.

Return to the Summary Table.

Compare Control controls the source for the compare registers

Table 14-6. RTICOMPCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
12	COMP3SEL	R/W	0h	COMPSEL 3: Compare Select 3. This bit determines the counter with which the compare value hold in compare register 3 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1
11-9	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
8	COMP2SEL	R/W	0h	COMPSEL 2: Compare Select 2. This bit determines the counter with which the compare value hold in compare register 2 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1
7-5	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect

Table 14-6. RTICOMPCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	COMP1SEL	R/W	0h	COMPSEL 1: Compare Select 1. This bit determines the counter with which the compare value hold in compare register 1 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1
3-1	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
0	COMP0SEL	R/W	0h	COMPSEL 0: Compare Select 0. This bit determines the counter with which the compare value hold in compare register 0 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1

ADVANCE INFORMATION

14.4.5 RTIFRC0 Register (Offset = 10h) [Reset = 00000000h]

RTIFRC0 is shown in Table 14-7.

Return to the Summary Table.

Free Running Counter 0 current value of free running counter 0

Table 14-7. RTIFRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FRC0	R/W	0h	FRC0: Free Running Counter 0. This registers holds the current value of the Free Running Counter 0 and will be updated continuously. User and privilege mode (read): current value of the counter Privilege mode (write): The counter can be preset by writing to this register. The counter increments then from this written value upwards. Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0.

14.4.6 RTIUC0 Register (Offset = 14h) [Reset = 00000000h]

RTIUC0 is shown in Table 14-8.

Return to the Summary Table.

Up Counter 0 current value of prescale counter 0

Table 14-8. RTIUC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UC0	R/W	0h	UC0: Up Counter 0. This registers 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. This gives effectively a 64 bit read of both counters, without having the problem of a counter being updated between two consecutive reads on Up Counter 0 and Free Running Counter 0. User and privilege mode (read): value of the counter when the Free Running Counter 0 was read Privilege mode (write): the counter can be preset by writing to this register. The counter increments then from this written value upwards. Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0. Note: Preset value concern 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.

14.4.7 RTICPUC0 Register (Offset = 18h) [Reset = 00000000h]

RTICPUC0 is shown in Table 14-9.

Return to the Summary Table.

Compare Up Counter 0 compare value compared with prescale counter 0

Table 14-9. RTICPUC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPUC0	R/W	0h	This registers holds the compare value, which is compared with the Up Counter 0. When the compare matches, Free Running counter 0 is incremented. The Up Counter is set to zero when the counter value matches the CPUC0 value. The value set in this prescales the RTI clock. If CPUC 0 = 0: then, frequency = RTICLK/ (2^{32}) If CPUC0 ≠ 0: then , frequency = RTICLK/(CPUC0 + 1) User and privilege mode (read): current compare value Privilege mode (write when TBEXT = 0): the compare value is updated Privilege mode (write when TBEXT = 1): the compare value is not changed

14.4.8 RTICAFRC0 Register (Offset = 20h) [Reset = 00000000h]

RTICAFRC0 is shown in Table 14-10.

Return to the Summary Table.

Capture Free Running Counter 0 current value of free running counter 0 on external event

Table 14-10. RTICAFRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAFRC0	R/W	0h	CAFRC0: Capture Free Running Counter 0. This registers captures the current value of the Free Running Counter 0 when a event occurs, controlled by the external capture control block. User and privilege mode (read): value of Free Running Counter 0 on a capture event

14.4.9 RTICAUC0 Register (Offset = 24h) [Reset = 00000000h]

RTICAUC0 is shown in Table 14-11.

Return to the Summary Table.

Capture Up Counter 0 current value of prescale counter 0 on external event

Table 14-11. RTICAUC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAUC0	R/W	0h	CAUC0: Capture Up Counter 0. This registers captures the current value of the Up Counter 0 when a event occurs, controlled by the external capture control block. The read sequence has to be the same as with Up Counter 0 and Free Running Counter 0. So the RTICAFRC0 register has to be read first, 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. User and privilege mode (read): value of Up Counter 0 on a capture event

14.4.10 RTIFRC1 Register (Offset = 30h) [Reset = 00000000h]

RTIFRC1 is shown in Table 14-12.

Return to the Summary Table.

Free Running Counter 1 current value of free running counter 1

Table 14-12. RTIFRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FRC1	R/W	0h	FRC1: Free Running Counter 1. This registers holds the current value of the Free Running Counter 1 and will be updated continuously. User and privilege mode (read): current value of the counter Privilege mode (write): The counter can be preset by writing to this register. The counter increments then from this written value upwards. Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC1 and RTIFRC1.

14.4.11 RTIUC1 Register (Offset = 34h) [Reset = 00000000h]

RTIUC1 is shown in Table 14-13.

Return to the Summary Table.

Up Counter 1 current value of prescale counter 1

Table 14-13. RTIUC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UC1	R/W	0h	UC1: Up Counter 1. This registers 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. This gives effectively a 64 bit read of both counters, without having the problem of a counter being updated between two consecutive reads on Up Counter 1 and Free Running Counter 1. User and privilege mode (read): value of the counter when the Free Running Counter 1 was read Privilege mode (write): the counter can be preset by writing to this register. The counter increments then from this written value upwards. Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC1 and RTIFRC1. Note: Preset value concern 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.

14.4.12 RTICPUC1 Register (Offset = 38h) [Reset = 00000000h]

RTICPUC1 is shown in Table 14-14.

Return to the Summary Table.

Compare Up Counter 1 compare value compared with prescale counter 1

Table 14-14. RTICPUC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CPUC1	R/W	0h	This registers holds the compare value, which is compared with the Up Counter 1. When the compare matches, Free Running Counter 1 is incremented. The Up Counter is set to zero when the counter value matches the CPUC1 value. The value set in this prescales the RTI clock. If CPUC 1 = 0: then, frequency = RTICLK/ (2^{32}) If CPUC1 ≠ 0: then , frequency = RTICLK/(CPUC1 + 1) User and privilege mode (read): current compare value Privilege mode (write when TBEXT = 0): the compare value is updated Privilege mode (write when TBEXT = 1): the compare value is not changed

14.4.13 RTICAFRC1 Register (Offset = 40h) [Reset = 00000000h]

RTICAFRC1 is shown in Table 14-15.

Return to the Summary Table.

Capture Free Running Counter 1 current value of free running counter 1 on external event

Table 14-15. RTICAFRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAFRC1	R/W	0h	CAFRC1: Capture Free Running Counter 1. This registers captures the current value of the Free Running Counter 1 when a event occurs, controlled by the external capture control block. User and privilege mode (read): value of Free Running Counter 1 on a capture event

14.4.14 RTICAUC1 Register (Offset = 44h) [Reset = 00000000h]

RTICAUC1 is shown in Table 14-16.

Return to the Summary Table.

Capture Up Counter 1 current value of prescale counter 1 on external event

Table 14-16. RTICAUC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAUC1	R/W	0h	CAUC1: Capture Up Counter 1. This registers captures the current value of the Up Counter 1 when a event occurs, controlled by the external capture control block. The read sequence has to be the same as with Up Counter 1 and Free Running Counter 1. So the RTICAFRC1 register has to be read first, 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. User and privilege mode (read): value of Up Counter 1 on a capture event

14.4.15 RTICOMP0 Register (Offset = 50h) [Reset = 00000000h]

RTICOMP0 is shown in Table 14-17.

Return to the Summary Table.

Compare 0 compare value to be compared with the counters

Table 14-17. RTICOMP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP0	R/W	0h	COMP0: Compare 0. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

14.4.16 RTIUDCP0 Register (Offset = 54h) [Reset = 00000000h]

RTIUDCP0 is shown in Table 14-18.

Return to the Summary Table.

Update Compare 0 value to be added to the compare register 0 value on compare match

Table 14-18. RTIUDCP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UDCP0	R/W	0h	UDCP0: Update Compare 0 Register. This registers holds a value, which is added to the value in the compare 0 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare 0 register on the next compare match Privilege mode (write): new update value

14.4.17 RTICOMP1 Register (Offset = 58h) [Reset = 00000000h]

RTICOMP1 is shown in Table 14-19.

Return to the Summary Table.

Compare 1 compare value to be compared with the counters

Table 14-19. RTICOMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP1	R/W	0h	COMP1: compare1. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

14.4.18 RTIUDCP1 Register (Offset = 5Ch) [Reset = 00000000h]

RTIUDCP1 is shown in Table 14-20.

Return to the Summary Table.

Update Compare 1 value to be added to the compare register 1 value on compare match

Table 14-20. RTIUDCP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UDCP1	R/W	0h	UDCP1: Update compare1 Register. This registers holds a value, which is added to the value in the compare1 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare1 register on the next compare match Privilege mode (write): new update value

14.4.19 RTICOMP2 Register (Offset = 60h) [Reset = 00000000h]

RTICOMP2 is shown in Table 14-21.

Return to the Summary Table.

Compare 2 compare value to be compared with the counters

Table 14-21. RTICOMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP2	R/W	0h	COMP2: compare 2. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

14.4.20 RTIUDCP2 Register (Offset = 64h) [Reset = 00000000h]

RTIUDCP2 is shown in Table 14-22.

Return to the Summary Table.

Update Compare 2 value to be added to the compare register 2 value on compare match

Table 14-22. RTIUDCP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UDCP2	R/W	0h	UDCP2: Update compare 2 Register. This registers holds a value, which is added to the value in the compare 2 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare 2 register on the next compare match Privilege mode (write): new update value

14.4.21 RTICOMP3 Register (Offset = 68h) [Reset = 00000000h]

RTICOMP3 is shown in Table 14-23.

Return to the Summary Table.

Compare 3 compare value to be compared with the counters

Table 14-23. RTICOMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP3	R/W	0h	COMP3: compare 3. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

14.4.22 RTIUDCP3 Register (Offset = 6Ch) [Reset = 00000000h]

RTIUDCP3 is shown in Table 14-24.

Return to the Summary Table.

Update Compare 3 value to be added to the compare register 3 value on compare match

Table 14-24. RTIUDCP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UDCP3	R/W	0h	UDCP3: Update compare 3 Register. This registers holds a value, which is added to the value in the compare 3 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare 3 register on the next compare match Privilege mode (write): new update value

14.4.23 RTITBLCOMP Register (Offset = 70h) [Reset = 00000000h]

RTITBLCOMP is shown in Table 14-25.

Return to the Summary Table.

Timebase Low Compare compare value to activate edge detection circuit

Table 14-25. RTITBLCOMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TBLCOMP	R/W	0h	TBLCOMP: Timebase Low Compare Value. This value determines when the edge detection circuit starts monitoring the NTUx signal. It will be compared with Up Counter 0. User and privilege mode (read): current compare value Privilege mode (write when TBEXT = 0): the compare value is updated Privilege mode (write when TBEXT = 1): the compare value is not changed Note: Reset behavior A reset does not generate a compare match.

14.4.24 RTITBHCOMP Register (Offset = 74h) [Reset = 00000000h]

RTITBHCOMP is shown in Table 14-26.

Return to the Summary Table.

Timebase High Compare compare value to deactivate edge detection circuit

Table 14-26. RTITBHCOMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TBHCOMP	R/W	0h	TBHCOMP: Timebase High Compare Value. This value determines when the edge detection circuit will stop monitoring the NTUx signal. It will be compared with Up Counter 0. RTITBHCOMP has to be less than RTICPUC0, since RTIUC0 will be reset when RTICPUC0 is reached. Example: The NTUx edge detection circuit should be active +/- 10 RTICLK cycles around RTICPUC0. RTICPUC 0 = 0x00000050 RTITBLCOMP = 0x000046 RTITBHCOMP = 0x00000009 User and privilege mode (read): current compare value Privilege mode (write when TBEXT = 0): the compare value is updated Privilege mode (write when TBEXT = 1): the compare value is not changed Note: Reset behavior A reset does not generate a compare match.

14.4.25 RTISETINT Register (Offset = 80h) [Reset = 00000000h]

RTISETINT is shown in Table 14-27.

Return to the Summary Table.

Set Interrupt Enable sets interrupt enable bits int RTIINTCTRL without having to do a read-modify-write operation

Table 14-27. RTISETINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
18	SETOVL1INT	R/W	0h	SETOVL1INT: Set Free Running Counter 1 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt

Table 14-27. RTISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	SETOVL0INT	R/W	0h	SETOVL0INT: Set Free Running Counter 0 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
16	SETTBINT	R/W	0h	SETTBINT: Set Timebase Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
15-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11	SETDMA3	R/W	0h	SETDMA 3: Set Compare DMA Request 3. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
10	SETDMA2	R/W	0h	SETDMA 2: Set Compare DMA Request 2. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
9	SETDMA1	R/W	0h	SETDMA 1: Set Compare DMA Request 1. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
8	SETDMA0	R/W	0h	SETDMA 0: Set Compare DMA Request 0. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
7-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3	SETINT3	R/W	0h	SETINT 3: Set Compare Interrupt 3. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged
2	SETINT2	R/W	0h	SETINT 2: Set Compare Interrupt 2. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt

ADVANCE INFORMATION

Table 14-27. RTISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SETINT1	R/W	0h	SETINT 1: Set Compare Interrupt 1. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
0	SETINT0	R/W	0h	SETINT 0: Set Compare Interrupt 0. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt

14.4.26 RTICLEARINT Register (Offset = 84h) [Reset = 00000000h]

RTICLEARINT is shown in Table 14-28.

Return to the Summary Table.

Clear Interrupt Enable clears interrupt enable bits int RTIINTCTRL without having to do a read-modify-write operation

Table 14-28. RTICLEARINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
18	CLEAROVL1INT	R/W	0h	CLEAROVL1INT: CLEAR Free Running Counter 1 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
17	CLEAROVL0INT	R/W	0h	CLEAROVL0INT: CLEAR Free Running Counter 0 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
16	CLEARTBINT	R/W	0h	CLEARTBINT: CLEAR Timebase Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
15-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11	CLEARDMA3	R/W	0h	CLEARDMA 3: CLEAR Compare DMA Request 3. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request

Table 14-28. RTICLEARINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	CLEARDMA2	R/W	0h	CLEARDMA 2: CLEAR Compare DMA Request 2. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
9	CLEARDMA1	R/W	0h	CLEARDMA 1: CLEAR Compare DMA Request 1. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
8	CLEARDMA0	R/W	0h	CLEARDMA 0: CLEAR Compare DMA Request 0. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
7-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3	CLEARINT3	R/W	0h	CLEARINT 3: CLEAR Compare Interrupt 3. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
2	CLEARINT2	R/W	0h	CLEARINT 2: CLEAR Compare Interrupt 2. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
1	CLEARINT1	R/W	0h	CLEARINT 1: CLEAR Compare Interrupt 1. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
0	CLEARINT0	R/W	0h	CLEARINT 0: CLEAR Compare Interrupt 0. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt

ADVANCE INFORMATION

14.4.27 RTIINTFLAG Register (Offset = 88h) [Reset = 00000000h]

RTIINTFLAG is shown in Table 14-29.

Return to the Summary Table.

Interrupt Flags interrupt pending bits

Table 14-29. RTIINTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
18	OVL1INT	R/W	0h	OVL1INT: Free Running Counter 1 Overflow Interrupt Flag. User and privilege mode (read): determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
17	OVL0INT	R/W	0h	OVL0INT: Free Running Counter 0 Overflow Interrupt Flag. User and privilege mode (read): determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
16	TBINT	R/W	0h	User and privilege mode (read): this flag is set when the TBEXT bit is cleared by detection of a missing external clockedge. It will not be set by clearing TBEXT by software. determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
15-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3	INT3	R/W	0h	INT 3: Interrupt Flag 3. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
2	INT2	R/W	0h	INT 2: Interrupt Flag 2. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
1	INT1	R/W	0h	INT 1: Interrupt Flag 1. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
0	INT0	R/W	0h	INT 0: Interrupt Flag 0. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0

14.4.28 RTIDWDCTRL Register (Offset = 90h) [Reset = 00000000h]

RTIDWDCTRL is shown in Table 14-30.

Return to the Summary Table.

Digital Watchdog Control Enables the Digital Watchdog

Table 14-30. RTIDWDCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DWDCTRL	R/W	0h	DWDCTRL: Digital Watchdog Control. User and priviledge mode (read): 0x5312ACED = DWD counter is disabled. This is the default value. 0xA98559DA = DWD counter is enabled Any other value = DWD counter state is unchanged (enabled or disabled) Priviledge mode (write): 0xA98559DA = DWD counter is enabled Any other value = State of DWD counter is unchanged (stays enabled or disabled) Note: One-Write Functionality of DWDCTRL Register The RTIDWDCTRL register implements a one-write functionality, such that the application cannot write to this registermore than once. Writing the default value will not enable the watchdog as described above. Writing the enable value will start the watchdog counters. A write to RTIDWDCTRL will only be enabled after a system reset again.

ADVANCE INFORMATION

14.4.29 RTIDWDPRLD Register (Offset = 94h) [Reset = 00000000h]

RTIDWDPRLD is shown in Table 14-31.

Return to the Summary Table.

Digital Watchdog Preload sets the experation time of the Digital Watchdog

Table 14-31. RTIDWDPRLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11-0	DWDPRLD	R/W	0h	DWDPRLD: Digital Watchdog Preload Value. User and priviledge mode (read): A read from this register in any CPU mode returns the current preload value. Priviledge mode (write): If the DWD is always enabled after reset is released: The DWD starts counting down from the reset value of the counter, that is, 0x002DFFFF. The application can configure the DWD preload register any time before this down counter expires. When the application services the DWD, the preload register contents are copied left-justified into the DWD down counter and it starts counting down from that value. If the DWD is implemented such that the down counter is enabled by software: 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 = (RTIDWDPRLD+1) x 2^{13} / RTICLK1 where: RTIDWDPRLD = 0...4095

14.4.30 RTIWDSTATUS Register (Offset = 98h) [Reset = 00000000h]

RTIWDSTATUS is shown in Table 14-32.

Return to the Summary Table.

Watchdog Status reflects the status of Analog and Digital Watchdog

Table 14-32. RTIWDSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect

Table 14-32. RTIWDSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DWWD_ST	R/W	0h	DWWD ST: Windowed Watchdog Status. This bit denotes whether the time-window defined by the windowed watchdog configuration has been violated, or if a wrong key or key sequence was written to service the watchdog. User and priviledge mode (read): 0 = no time-window violation has occurred. 1 = a time-window violation has occurred. The watchdog will generate either a system reset or a non-maskable interrupt to the CPU in this case. Priviledge mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0. This will also clear all other status flags in the RTIWDSTATUS register except for the AWD ST flag. Clearing of the status flags will deassert the non-maskable interrupt generated due to violation of the DWWD.
4	ENDTIMEVIOL	R/W	0h	END TIME VIOL: Windowed Watchdog End Time Violation Status. This bit denotes whether the end-time defined by the windowed watchdog configuration has been violated. This bit is effectively a copy of the DWD ST status flag. User and priviledge mode (read): 0 = no end-time window violation has occurred. 1 = the end-time defined by the windowed watchdog configuration has been violated. Priviledge mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0.
3	STARTTIMEVIOL	R/W	0h	START TIME VIOL: Windowed Watchdog Start Time Violation Status. This bit denotes whether the start-time defined by the windowed watchdog configuration has been violated. This indicates that the WWD was serviced before the service window was opened. User and priviledge mode (read): 0 = no start-time window violation has occurred. 1 = the start-time defined by the windowed watchdog configuration has been violated. Priviledge mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0.
2	KEYST	R/W	0h	KEYST: Watchdog KeyStatus. This bit denotes a reset generated by a wrong key or a wrong key-sequence written to the RTIWDKEY register. User and priviledge mode (read): 0 = no wrong key or key-sequence written 1 = wrong key or key-sequence written to RTIWDKEY register Priviledge mode (write): 0 = leaves the current value unchanged 1 = clears the bit to 0
1	DWDST	R/W	0h	DWDST: Digital Watchdog Status. This bit is effectively a copy of the END TIME VIOL status flag and is maintained for compatibility reasons. User and priviledge mode (read): 0 = DWD timeout period not expired 1 = DWD timeout period has expired Priviledge mode (write): 0 = leaves the current value unchanged 1 = clears the bit to 0
0	AWDST	R/W	0h	AWDST: Analog Watchdog Status. User and priviledge mode (read): 0 = AWD pin 0 –> 1 threshold not exceeded 1 = AWD pin 0 –> 1 threshold exceeded Priviledge mode (write): 0 = leaves the current value unchanged 1 = clears the bit to 0

ADVANCE INFORMATION

14.4.31 RTIWDKEY Register (Offset = 9Ch) [Reset = 00000000h]

RTIWDKEY is shown in Table 14-33.

Return to the Summary Table.

Watchdog Key correct written key values discharge the external capacitor

Table 14-33. RTIWDKEY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
15-0	WDKEY	R/W	0h	WDKEY: Watchdog Key. User and privilege mode reads are indeterminate. Privilege mode (write): A write of 0xE51A followed by 0xA35C in two separate write operations defines the Key Sequence and discharges the watchdog capacitor. This also causes the upper 12 bits of the DWD down counter to be reloaded with the contents of the DWD preload register and the lower 13 bits to become all 1's. Writing any other value causes a digital watchdog reset, as shown in Table 1-3. Note: Register write access time precaution The user has to take into account that the write to the register takes 3 VCLK cycle. This needs to be considered for the AWD/DWD expiration calculation.

14.4.32 RTIDWDCNTR Register (Offset = A0h) [Reset = 00000000h]

RTIDWDCNTR is shown in Table 14-34.

Return to the Summary Table.

Digital Watchdog Down Counter current value of DWD down counter

Table 14-34. RTIDWDCNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
24-0	DWDCNTR	R/W	0h	DWDCNTR: Digital Watchdog Down Counter. The value of the DWDCNTR after a system reset is 0x002D_FFFF. When the DWD is enabled and the DWD counter starts counting down from this value with an RTICLK1 time base of 3MHz, a watchdog reset will be generated in 1 second. User and privilege mode (read): Reads return the current counter value. Privilege mode (write): Writes don't have an effect.

14.4.33 RTIWWDRXNCTRL Register (Offset = A4h) [Reset = 00000000h]

RTIWWDRXNCTRL is shown in Table 14-35.

Return to the Summary Table.

Windowed Watchdog Reaction Control configures the windowed watchdog to either generate a non-maskable interrupt to the CPU or to generate a system reset

Table 14-35. RTIWWDRXNCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect

Table 14-35. RTIWWDRXNCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	WWDRXN	R/W	0h	WWDRXN: Digital Windowed Watchdog Reaction. User and privilege mode (read), privileged mode (write): 0x5 = This is the default value. 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. 0xA = 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. Writing any other value will cause a system reset if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all. Note: Configuration of DWWD Reaction 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.

14.4.34 RTIWWDSIZECTRL Register (Offset = A8h) [Reset = 00000000h]

RTIWWDSIZECTRL is shown in Table 14-36.

Return to the Summary Table.

Windowed Watchdog Size Control configures the size of the window for the digital windowed watchdog

Table 14-36. RTIWWDSIZECTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WWDSIZE	R/W	0h	WWDSIZE: Digital Windowed Watchdog Window Size. User and privilege mode (read), privileged mode (write): Value written to WWDSIZE Window Size 0x00000005 100% (Functionality same as the time-out digital watchdog.) 0x00000050 50% 0x00000500 25% 0x00005000 12.5% 0x00050000 6.25% 0x00500000 3.125% Any other value 3.125% Note: Incorrect value being written to watchdog window size control register If an incorerct value is written to the WWDSIZE field, or if a system disturbance causes the WWDSIZE field to have a value other than 0x5, 0x50, 0x500, 0x5000, 0x50000, or 0x500000, then the window size will be configured to be 3.125%. This increases the chances of getting a reset due to the windowed watchdog, which enables the system to handle the cause for the incorrect configuration. Note: Configuration of DWWD Window Size 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

14.4.35 RTIINTCLRENABLE Register (Offset = ACh) [Reset = 00000000h]

RTIINTCLRENABLE is shown in Table 14-37.

Return to the Summary Table.

RTI Compare Interrupt Clear Enable enable the auto clear functionality for each of the compare interrupts

Table 14-37. RTIINTCLRENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
27-24	INTCLRENABLE3	R/W	0h	INTCLRENABLE3. Enables the auto-clear functionality on the compare 3 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 3 interrupt is disabled. Any other value = Auto-clear for compare 3 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 3 interrupt. Any other value = Enables the auto-clear functionality on the compare 3 interrupt.
23-20	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
19-16	INTCLRENABLE2	R/W	0h	INTCLRENABLE2. Enables the auto-clear functionality on the compare 2 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 2 interrupt is disabled. Any other value = Auto-clear for compare 2 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 2 interrupt. Any other value = Enables the auto-clear functionality on the compare 2 interrupt.
15-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11-8	INTCLRENABLE1	R/W	0h	INTCLRENABLE1. Enables the auto-clear functionality on the compare 1 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 1 interrupt is disabled. Any other value = Auto-clear for compare 1 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 1 interrupt. Any other value = Enables the auto-clear functionality on the compare 1 interrupt.
7-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3-0	INTCLRENABLE0	R/W	0h	INTCLRENABLE0. Enables the auto-clear functionality on the compare 0 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 0 interrupt is disabled. Any other value = Auto-clear for compare 0 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 0 interrupt. Any other value = Enables the auto-clear functionality on the compare 0 interrupt.

14.4.36 RTICOMP0CLR Register (Offset = B0h) [Reset = 00000000h]

RTICOMP0CLR is shown in Table 14-38.

Return to the Summary Table.

Compare 0 Clear compare value to be compared with the counter to clear the compare0 interrupt line

Table 14-38. RTICOMP0CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP0CLR	R/W	0h	COMP0CLR: Compare 0 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the compare 0 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

14.4.37 RTICOMP1CLR Register (Offset = B4h) [Reset = 00000000h]

RTICOMP1CLR is shown in Table 14-39.

Return to the Summary Table.

Compare 1 Clear compare value to be compared with the counter to clear the compare1 interrupt line

Table 14-39. RTICOMP1CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP1CLR	R/W	0h	COMP1CLR: Compare 1 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the Compare 1 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

14.4.38 RTICOMP2CLR Register (Offset = B8h) [Reset = 00000000h]

RTICOMP2CLR is shown in Table 14-40.

Return to the Summary Table.

Compare 2 Clear compare value to be compared with the counter to clear the compare2 interrupt line

Table 14-40. RTICOMP2CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP2CLR	R/W	0h	COMP2CLR: Compare 2 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the Compare 2 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

14.4.39 RTICOMP3CLR Register (Offset = BCh) [Reset = 00000000h]

RTICOMP3CLR is shown in Table 14-41.

Return to the Summary Table.

Compare 3 Clear compare value to be compared with the counter to clear the compare3 interrupt line

Table 14-41. RTICOMP3CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMP3CLR	R/W	0h	COMP3CLR: Compare 3 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the Compare 3 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

15 UART/SCI

This chapter contains the description of the serial communication interface (SCI) module.

15.1 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.

15.1.1 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 communication mode 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.

15.1.2 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 VCLK.

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 can 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 15-1 shows the detailed SCI block diagram.

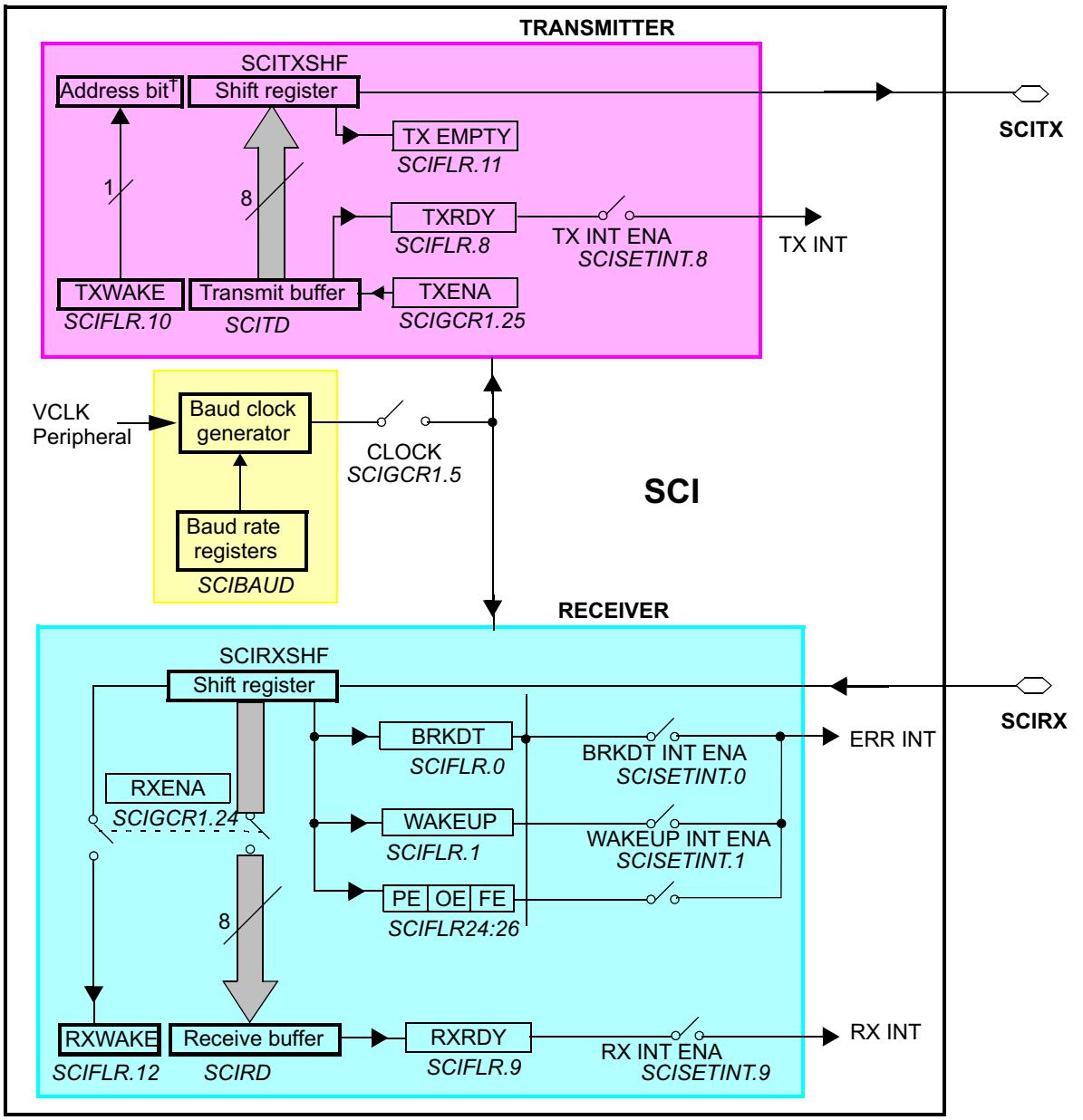


Figure 15-1. Detailed SCI Block Diagram

ADVANCE INFORMATION

15.2 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

15.2.1 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 15-2.

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 15-2 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 15-2 use one stop bit per frame.

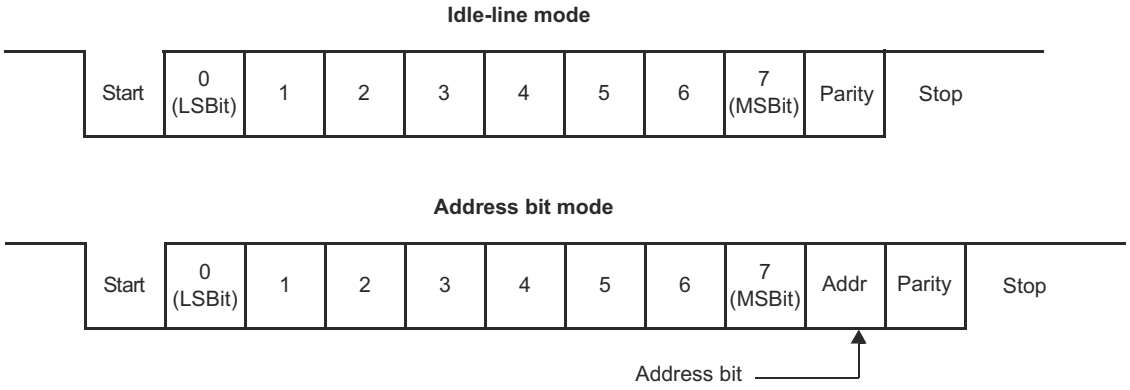


Figure 15-2. Typical SCI Data Frame Formats

15.2.2 SCI Timing Mode

The SCI can be configured to use asynchronous or isosynchronous timing using TIMING MODE bit in SCIGCR1 register.

15.2.2.1 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 15-3 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.

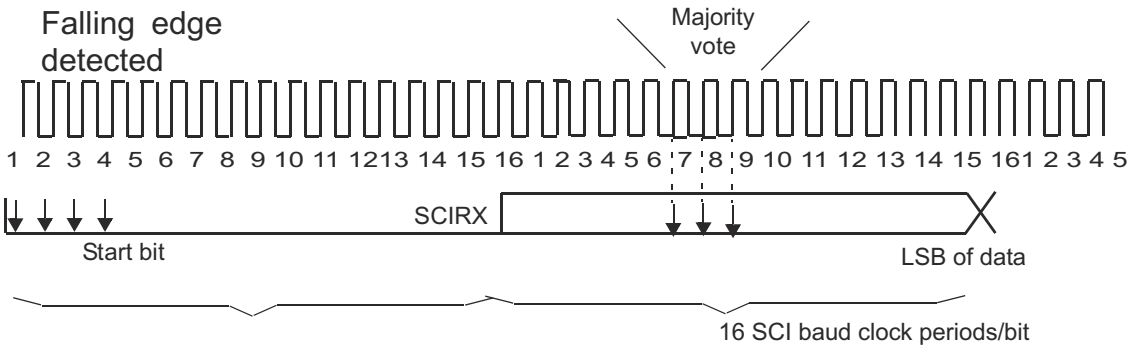


Figure 15-3. Asynchronous Communication Bit Timing

15.2.2.2 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.**

ADVANCE INFORMATION

15.2.3 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 of the BRS register to select the required baud rates.

In asynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$\textit{Asynchronous baud value} = \frac{\text{VCLK Frequency}}{16 * (\text{BAUD} + 1)}$$

For BAUD = 0,

$$\textit{Asynchronous baud value} = \frac{\text{VCLK Frequency}}{32} \quad (4)$$

In isosynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$\textit{Isosynchronous baud value} = \frac{\text{VCLK Frequency}}{\text{BAUD} + 1}$$

For BAUD = 0,

$$\textit{Isosynchronous baud value} = \frac{\text{VCLK Frequency}}{32} \quad (5)$$

15.2.4 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..

15.2.4.1 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 15-4 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:

Step 1 : Write a 1 to the TXWAKE bit.

Step 2 : 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.

Step 3 : Wait for the SCI to clear the TXWAKE flag.

Step 4 : 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.

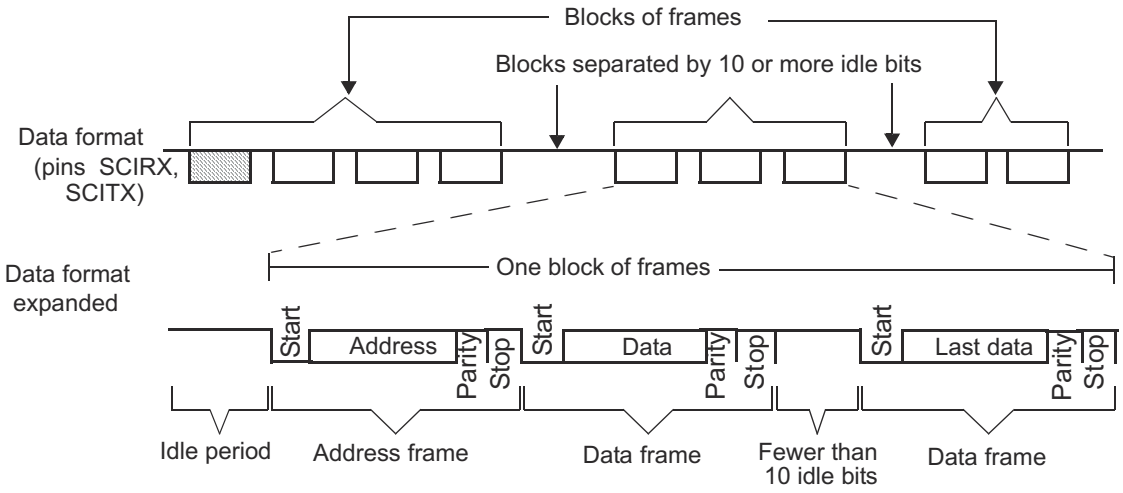


Figure 15-4. Idle-Line Multiprocessor Communication Format

15.2.4.2 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 15-5 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.

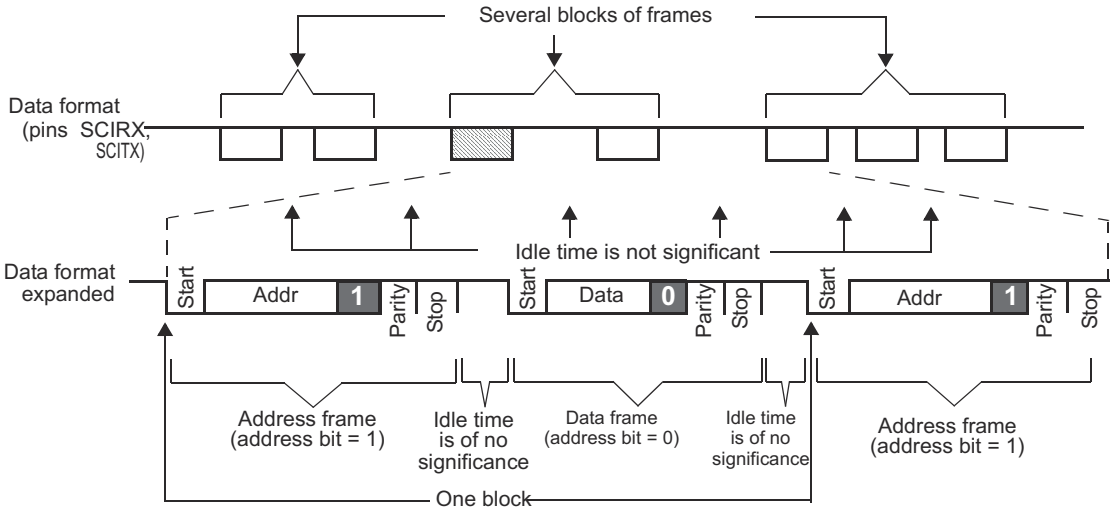


Figure 15-5. Address-Bit Multiprocessor Communication Format

ADVANCE INFORMATION

15.3 SCI Interrupts

The SCI module has two interrupt lines, level 0 and level 1, to the interrupt manager (NVIC/IM) module (see Figure 15-6). 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 and 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.

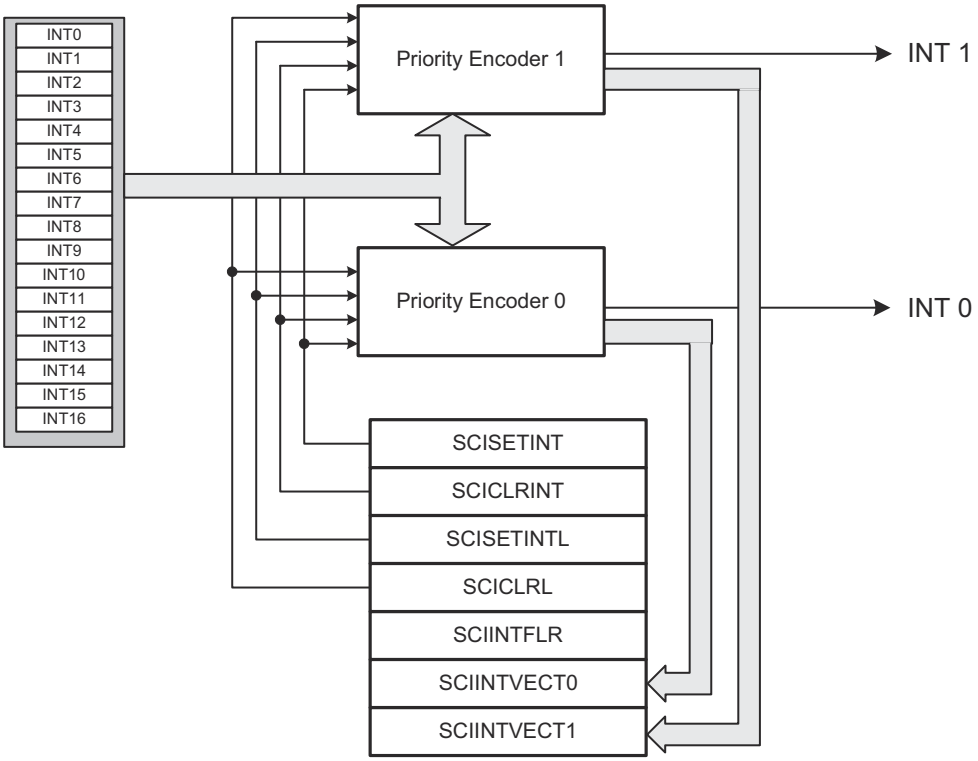


Figure 15-6. General Interrupt Scheme

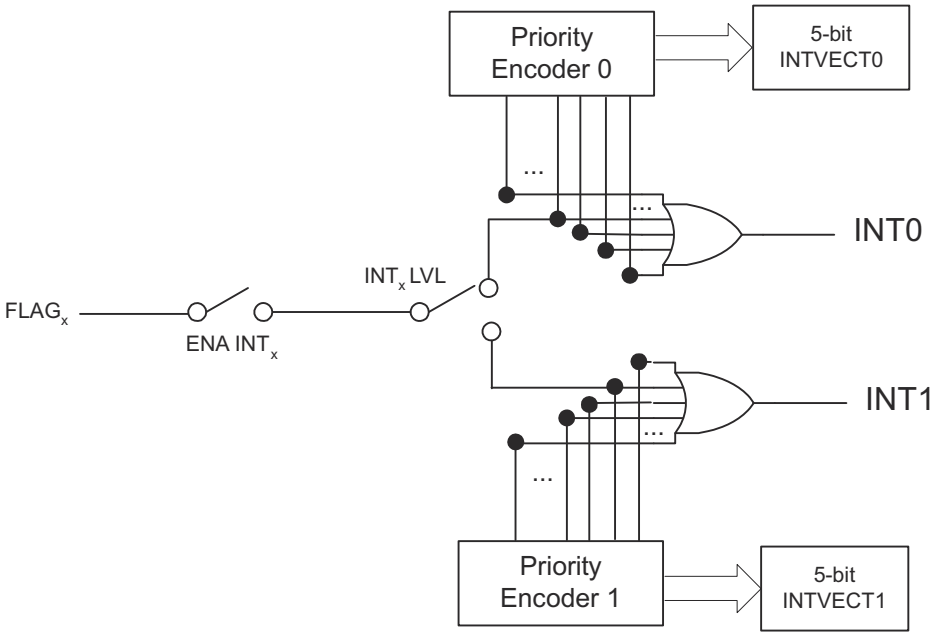


Figure 15-7. Interrupt Generation for Given Flags

15.3.1 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.

15.3.2 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.

15.3.3 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 interrupt is triggered once WAKEUP flag is set.

15.3.4 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 the following 7 interrupts as listed in Table 15-1.

Table 15-1. 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.

15.4 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.

15.4.1 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, wakeup, 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 15-2 specifies the bit values for DMA requests in multiprocessor modes.

Table 15-2. 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

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.

15.4.2 Transmit DMA Requests

DMA functionality is enabled and disabled by the CPU with the SET TX DMA and 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. The DMA will write the first byte to the transmit buffer.

15.5 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 in the SCIGCR0 register is set to 1. Of particular importance is the SWnRST bit in the SCIGCR1 register. The SWnRST is an active-low bit 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 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 the SWnRST bit is cleared to 0 the entire time that the SCI is being configured, the order in which the registers are programmed is not important.

- Enable SCI by setting the RESET bit to 1.
- Clear the SWnRST bit to 0 before SCI is configured.
- Select the desired frame format by programming the SCIGCR1 register.
- Set both the RX FUNC and TX FUNC bits in SCIPIO0 to 1 to configure the SCIRX and SCITX pins for SCI functionality.
- Select the baud rate to be used for communication by programming the BRS register.
- Set the CLOCK bit in SCIGCR1 to 1 to select the internal clock.
- Set the CONT bit in SCIGCR1 to 1 to make SCI not 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 in SCIGCR1 to 1 to connect the transmitter to the receiver internally (this feature is used to perform a self-test).
- Set the RXENA bit in SCIGCR1 to 1, if data is to be received.
- Set the TXENA bitin SCIGCR1 to 1, if data is to be transmitted.
- Set the SWnRST bit to 1 after SCI is configured.
- Perform receiving or transmitting data (see Section 15.5.1 and Section 15.5.2).

15.5.1 Receiving Data

The SCI receiver is enabled to receive messages if both 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 the FE, OE, or PE flags if any of these error conditions were detected in the received data. These error conditions are supported with configurable interrupt capability. The wakeup 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.

You can receive data by:

1. Polling Receive Ready Flag
2. Receive Interrupt
3. DMA

In polling method, software can poll for the RXRDY bit and read the data from SCIRD register once RXRDY is set high. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use either the interrupt or DMA method. To use the interrupt method, the SET RX INT bit is set. To use the DMA method, the SET RX DMA bit is set. Either an interrupt or a DMA request is generated the moment the RXRDY bit is set.

15.5.2 Transmitting Data

The SCI transmitter is enabled if both 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 the data. The TXRDY and TX EMPTY bits 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.

You can transmit data by:

1. Polling Transmit Ready Flag
2. Transmit Interrupt
3. DMA

In polling method, software can poll for the TXRDY bit to go high before writing the data to the SCITD register. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use either the interrupt or DMA method. To use the interrupt method, the SET TX INT bit is set. To use the DMA method, the SET TX DMA bit is set. Either an interrupt or a DMA request is generated the moment the TXRDY bit 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 either be done by 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.

15.6 APP_UART Registers

Table 15-3 lists the memory-mapped registers for the APP_UART registers. All register offset addresses not listed in Table 15-3 should be considered as reserved locations and the register contents should not be modified.

Table 15-3. APP_UART Registers

Offset	Acronym	Register Name	Section
0h	SCIGCR0	The SCIGCR0 register defines the module reset	Go
4h	SCIGCR1	The SCIGCR1 register defines the frame format, protocol, and communication mode used by the SCI	Go
8h	RESERVED1	Reserved	Go
Ch	SCISETINT	SCI Set Interrupt Register	Go
10h	SCICLEARINT	SCI Clear Interrupt Register	Go
14h	SCISETINTLVL	SCI Set Interrupt Level Register	Go
18h	SCICLEARINTLVL	SCI Clear Interrupt Level Register	Go
1Ch	SCIFLR	SCI Flags Register	Go
20h	SCIINTVECT0	SCI Interrupt Offset Vector 0 Register	Go
24h	SCIINTVECT1	SCI Interrupt Offset Vector 1 Register	Go
28h	SCICHAR	SCI Character Control Register	Go
2Ch	SCIBAUD	SCI Baud Rate Selection Register	Go
30h	SCIED	Receiver Emulation Data Buffer	Go
34h	SCIRD	Receiver Data Buffer	Go
38h	SCITD	Transmit Data Buffer Register	Go
3Ch	SCIPIO0	SCI Pin I/O Control Register 0	Go
40h	SCIPIO1	SCI Pin I/O Control Register 1	Go
44h	SCIPIO2	SCI Pin I/O Control Register 2	Go
48h	SCIPIO3	SCI Pin I/O Control Register 3	Go
4Ch	SCIPIO4	SCI Pin I/O Control Register 4	Go
50h	SCIPIO5	SCI Pin I/O Control Register 5	Go
54h	SCIPIO6	SCI Pin I/O Control Register 6	Go
58h	SCIPIO7	SCI Pin I/O Control Register 7	Go
5Ch	SCIPIO8	SCI Pin I/O Control Register 8	Go
60h	RESERVED2	Reserved	Go
64h	RESERVED3	Reserved	Go
68h	RESERVED4	Reserved	Go
6Ch	RESERVED5	Reserved	Go
70h	RESERVED6	Reserved	Go
74h	RESERVED7	Reserved	Go
78h	RESERVED8	Reserved	Go
7Ch	RESERVED9	Reserved	Go
80h	SCIPIO9	SCI Pin I/O Control Register 9	Go
90h	SCIIODCTRL	SCI IO DFT Control	Go

Complex bit access types are encoded to fit into small table cells. Table 15-4 shows the codes that are used for access types in this section.

Table 15-4. APP_UART Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		

Table 15-4. APP_UART Access Type Codes (continued)

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

15.6.1 SCIGCR0 Register (Offset = 0h) [Reset = 00000000h]

SCIGCR0 is shown in Table 15-5.

Return to the Summary Table.

The SCIGCR0 register defines the module reset

Table 15-5. SCIGCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	RESET	R/W	0h	GIO reset

15.6.2 SCIGCR1 Register (Offset = 4h) [Reset = 00000000h]

SCIGCR1 is shown in Table 15-6.

Return to the Summary Table.

The SCIGCR1 register defines the frame format, protocol, and communication mode used by the SCI

Table 15-6. SCIGCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	TXENA	R/W	0h	Data is transferred from SCITD to SCITXSHF only when the TXENA bit is set
24	RXENA	R/W	0h	Allows the receiver to transfer data from the shift buffer to the receive buffer
23-18	RESERVED	R	0h	Reserved
17	CONT	R/W	0h	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
16	LOOP_BACK	R/W	0h	Enable bit for loopback mode
15-10	RESERVED	R	0h	Reserved
9	POWERDOWN	R/W	0h	When the POWERDOWN bit is set, the SCI attempts to enter local low-power mode
8	SLEEP	R/W	0h	In a multiprocessor configuration, this bit controls the receive sleep function. Clearing this bit brings the SCI out of sleep mode
7	SW_nRESET	R/W	0h	Software reset (active low)
6	RESERVED	R	0h	Reserved
5	CLOCK	R/W	0h	SCI internal clock enable
4	STOP	R/W	0h	SCI number of stop bits
3	PARITY	R/W	0h	SCI parity odd/even selection
2	PARITY_ENA	R/W	0h	SCI parity enable
1	TIMING_MODE	R/W	0h	SCI timing mode bit (0=Isosynchronous timing, 1=Asynchronous timing)

Table 15-6. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	COMM_MODE	R/W	0h	SCI communication mode bit (0=Idle-line mode, 1=Address-bit mode)

15.6.3 RESERVED1 Register (Offset = 8h) [Reset = 00000000h]

RESERVED1 is shown in Table 15-7.

Return to the Summary Table.

Reserved

Table 15-7. RESERVED1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.4 SCISETINT Register (Offset = Ch) [Reset = 00000000h]

SCISETINT is shown in Table 15-8.

Return to the Summary Table.

SCI Set Interrupt Register

Table 15-8. SCISETINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	SET_FE_INT	R/W	0h	Set Framing-Error Interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
25	SET_OE_INT	R/W	0h	Set Overrun-Error Interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
24	SET_PE_INT	R/W	0h	Set Parity Interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
23-19	RESERVED	R	0h	Reserved
18	SET_RX_DMA_ALL	R/W	0h	Determines if a separate interrupt is generated for the address frames sent in multiprocessor communications User and privilege mode (read): 0 = DMA request is disabled for address frames (RX interrupt request is enabled for address frames) 1 = DMA request is enabled for address and data frames User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request for address and data frames
17	SET_RX_DMA	R/W	0h	To select receiver DMA requests, this bit must be set. If it is cleared, interrupt requests are generated depending on bit SCISETINT.9 User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request

Table 15-8. SCISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	SET_TX_DMA	R/W	0h	To select 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.8) User and privilege mode (read): 0 = TX interrupt request selected 1 = TX DMA request selected User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
15-10	RESERVED	R	0h	Reserved
9	SET_RX_INT	R/W	0h	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. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
8	SET_TX_INT	R/W	0h	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. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
7-2	RESERVED	R	0h	Reserved
1	SET_WAKEUP_INT	R/W	0h	Set Wakeup interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
0	SET_BRKDT_INT	R/W	0h	Set Break-detect interrupt. Setting this bit enables the SCI to generate an error interrupt if a break condition is detected on the SCIRX pin. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt

15.6.5 SCICLEARINT Register (Offset = 10h) [Reset = 00000000h]

SCICLEARINT is shown in Table 15-9.

Return to the Summary Table.

SCI Clear Interrupt Register

Table 15-9. SCICLEARINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	CLR_FE_INT	R/W	0h	Clear Framing-Error Interrupt: Setting this bit disables the SCI module to generate an interrupt when there is a Framing error. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt

Table 15-9. SCICLEARINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	CLR_OE_INT	R/W	0h	Clear Overrun-Error Interrupt. This bit disables the SCI overrun interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
24	CLR_PE_INT	R/W	0h	Clear Parity Interrupt. Setting this bit disables the SCI Parity error interrupt. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
23-19	RESERVED	R	0h	Reserved
18	CLR_RX_DMA_ALL	R/W	0h	User and privilege mode (read): 0 = DMA request is disabled for address frames (RX interrupt request is enabled for address frames). DMA request is enabled for data frames. 1 = DMA request is enabled for address and data frames User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request for address frames
17	CLR_RX_DMA	R/W	0h	Clear RX DMA request. This bit disalbes the receive DMA request when set. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
16	CLR_TX_DMA	R/W	0h	Clear TX DMA request. This bit disables the transmit DMA request when set. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
15-10	RESERVED	R	0h	Reserved
9	CLR_RX_INT	R/W	0h	Clear Receiver interrupt. This bit disables the receiver interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
8	CLR_TX_INT	R/W	0h	Clear Transmitter interrupt. This bit disables the transmitter interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
7-2	RESERVED	R	0h	Reserved
1	CLR_WAKEUP_INT	R/W	0h	Clear Wakeup interrupt. This bit disables the wakeup interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt

Table 15-9. SCICLEARINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CLR_BRKDT_INT	R/W	0h	Clear Break-detect interrupt. This bit disables the Break-detect interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt

15.6.6 SCISETINTLVL Register (Offset = 14h) [Reset = 00000000h]

SCISETINTLVL is shown in Table 15-10.

Return to the Summary Table.

SCI Set Interrupt Level Register

Table 15-10. SCISETINTLVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	SET_FE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
25	SET_OE_INT_LVL	R/W	0h	Clear Overrun-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
24	SET_PE_INT_LVL	R/W	0h	Clear Parity Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
23-19	RESERVED	R	0h	Reserved
18	SET_RX_DMA_ALL_INT_LVL	R/W	0h	User and privilege mode (read): 0 = RX interrupt request for address frames mapped to INT0 line. 1 = RX interrupt request for address frames mapped to INT1 line. User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
17-16	RESERVED	R	0h	Reserved
15	SET_INC_BR_INT_LVL	R/W	0h	Reserved
14-10	RESERVED	R	0h	Reserved
9	SET_RX_INT_LVL	R/W	0h	Clear Receiver interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1

Table 15-10. SCISETINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SET_TX_INT_LVL	R/W	0h	Clear Transmitter interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
7-2	RESERVED	R	0h	Reserved
1	SET_WAKEUP_INT_LVL	R/W	0h	Clear Wakeup interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
0	SET_BRKDT_INT_LVL	R/W	0h	Clear Break-detect interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1

15.6.7 SCICLEARINTLVL Register (Offset = 18h) [Reset = 00000000h]

SCICLEARINTLVL is shown in Table 15-11.

Return to the Summary Table.

SCI Clear Interrupt Level Register

Table 15-11. SCICLEARINTLVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	CLR_FE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
25	CLR_OE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
24	CLR_PE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
23-19	RESERVED	R	0h	Reserved

Table 15-11. SCICLEARINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	CLR_RX_DMA_ALL_INT_LVL	R/W	0h	Clear receive DMA ALL interrupt level. User and privilege mode (read): 0 = RX interrupt request for address frames is mapped to INT0 line. 1 = RX interrupt request for address frames is mapped to INT1 line. User and privilege mode (write): 0 = Leaves the corresponding bit unchanged. 1 = Reset interrupt level to line INT0.
17-16	RESERVED	R	0h	Reserved
15	CLR_INC_BR_INT_LVL	R/W	0h	
14-10	RESERVED	R	0h	Reserved
9	CLR_RX_INT_LVL	R/W	0h	Clear Receiver interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
8	CLR_TX_INT_LVL	R/W	0h	Clear Transmitter interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
7-2	RESERVED	R	0h	Reserved
1	CLR_WAKEUP_INT_LVL	R/W	0h	Clear Wakeup interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
0	CLR_BRKDT_INT_LVL	R/W	0h	Clear Break-detect interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0

ADVANCE INFORMATION

15.6.8 SCIFLR Register (Offset = 1Ch) [Reset = 00000904h]

SCIFLR is shown in Table 15-12.

Return to the Summary Table.

SCI Flags Register

Table 15-12. SCIFLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	FE	R/W	0h	SCI framing error flag Read: 0=No framing error detected 1=Framing error detected Write: 0=No effect 1=Clears this bit to 0
25	OE	R	0h	SCI overrun error flag This bit is set when the transfer of data from SCIRXSHF to SCIRD overwrites unread data already in SCIRD
24	PE	R	0h	SCI parity error flag. This bit is set when a parity error is detected in the received data

Table 15-12. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-13	RESERVED	R	0h	Reserved
12	RXWAKE	R	0h	Receiver wakeup detect flag. The SCI sets this bit to indicate that the data currently in SCIRD is an address
11	TX_EMPTY	R	1h	Transmitter empty flag. The value of this flag indicates the contents of the transmitter's buffer register (SCITD) and shift register (SCITXSHF)
10	TXWAKE	R/W	0h	SCI 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
9	RXRDY	R	0h	SCI 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.
8	TXRDY	R	1h	Transmitter buffer register ready flag. When set, this bit indicates that the transmit buffer register (SCITD) is ready to receive another character.
7-4	RESERVED	R	0h	Reserved
3	Bus_busy_flag	R	0h	This bit indicates whether the receiver is in the process of receiving a frame.
2	IDLE	R	1h	SCI receiver in idle state. While this bit is set, the SCI looks for an idle period to resynchronize itself with the bit stream.
1	WAKEUP	R	0h	Wakeup flag. This bit is set by the SCI when receiver or transmitter activity has taken the module out of power-down mode.
0	BRKDT	R	0h	SCI break-detect flag. This bit is set when the SCI detects a break condition on the SCIRX pin.

15.6.9 SCIINTVECT0 Register (Offset = 20h) [Reset = 00000000h]

SCIINTVECT0 is shown in Table 15-13.

Return to the Summary Table.

SCI Interrupt Offset Vector 0 Register

Table 15-13. SCIINTVECT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	INTVECT0	R	0h	Interrupt vector offset for INT0

15.6.10 SCIINTVECT1 Register (Offset = 24h) [Reset = 00000000h]

SCIINTVECT1 is shown in Table 15-14.

Return to the Summary Table.

SCI Interrupt Offset Vector 1 Register

Table 15-14. SCIINTVECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	INTVECT1	R	0h	Interrupt vector offset for INT1

15.6.11 SCICHAR Register (Offset = 28h) [Reset = 00000000h]

SCICHAR is shown in Table 15-15.

Return to the Summary Table.

SCI Character Control Register

Table 15-15. SCICHAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	CHAR	R/W	0h	Sets the SCI data length from 1 to 8 bits

15.6.12 SCIBAUD Register (Offset = 2Ch) [Reset = 00000000h]

SCIBAUD is shown in Table 15-16.

Return to the Summary Table.

SCI Baud Rate Selection Register

Table 15-16. SCIBAUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	BAUD	R/W	0h	SCI 24-bit baud selection

15.6.13 SCIED Register (Offset = 30h) [Reset = 00000000h]

SCIED is shown in Table 15-17.

Return to the Summary Table.

Receiver Emulation Data Buffer

Table 15-17. SCIED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	ED	R	0h	Receiver Emulation Data Buffer

15.6.14 SCIRD Register (Offset = 34h) [Reset = 00000000h]

SCIRD is shown in Table 15-18.

Return to the Summary Table.

Receiver Data Buffer

Table 15-18. SCIRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RD	R	0h	Contains received data.

15.6.15 SCITD Register (Offset = 38h) [Reset = 00000000h]

SCITD is shown in Table 15-19.

Return to the Summary Table.

Transmit Data Buffer Register

Table 15-19. SCITD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	TD	R/W	0h	Contains Data to be transmitted. This is pushed to SCITXSHF(shift register) when TXENA bit is set in SCRGCR1 register.

15.6.16 SCIPIO0 Register (Offset = 3Ch) [Reset = 00000000h]

SCIPIO0 is shown in Table 15-20.

Return to the Summary Table.

SCI Pin I/O Control Register 0

Table 15-20. SCIPIO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_FUNC	R/W	0h	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	R/W	0h	Determines the data direction on the SCIRX pin if it is configured with general-purpose I/O functionality (RX FUNC = 0). See Table 12 for bit values. 0=SCIRX is a general-purpose input pin. 1=SCIRX is a general-purpose output pin
0	CLK_FUNC	R/W	0h	Clock function. Defines the function of pin SCICLK. 0=SCICLK is a general-purpose digital I/O pin. 1=SCICLK is the SCI serial clock pin.

15.6.17 SCIPIO1 Register (Offset = 40h) [Reset = 00000000h]

SCIPIO1 is shown in Table 15-21.

Return to the Summary Table.

SCI Pin I/O Control Register 1

Table 15-21. SCIPIO1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DIR	R/W	0h	Determines the data direction on the SCITX pin if it is configured with general-purpose I/O functionality (TX FUNC = 0). See Table 11 for bit values. 0=SCITX is a general-purpose input pin. 1=SCITX is a general-purpose output pin
1	RX_DIR	R/W	0h	Determines the data direction on the SCIRX pin if it is configured with general-purpose I/O functionality (RX FUNC = 0). See Table 12 for bit values. 0=SCIRX is a general-purpose input pin. 1=SCIRX is a general-purpose output pin
0	CLK_DIR	R/W	0h	Clock data direction. Determines the data direction on the SCICLK pin. The direction is defined differently depending upon the value of the CLK FUNC bit 0=SCICLK is a general-purpose input pin. 1=SCICLK is a general-purpose output pin

15.6.18 SCIPIO2 Register (Offset = 44h) [Reset = 00000000h]

SCIPIO2 is shown in Table 15-22.

Return to the Summary Table.

SCI Pin I/O Control Register 2

Table 15-22. SCIPIO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DATA_IN	R/W	0h	Contains current value on the SCITX pin. 0=SCITX value is logic low. 1=SCITX value is logic high.
1	RX_DATA_IN	R/W	0h	Contains current value on the SCIRX pin. 0=SCIRX value is logic low. 1=SCIRX value is logic high.
0	CLK_DATA_IN	R/W	0h	Contains the current value on pin SCICLK. 0=Pin SCICLK value is logic low. 1=Pin SCICLK value is logic high.

15.6.19 SCIPIO3 Register (Offset = 48h) [Reset = 00000000h]

SCIPIO3 is shown in Table 15-23.

Return to the Summary Table.

SCI Pin I/O Control Register 3

Table 15-23. SCIPIO3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DATA_OUT	R/W	0h	Contains the data to be output on pin SCITX if the following conditions are met: TX FUNC = 0 (SCITX pin is a general-purpose I/O.) TX DATA DIR = 1 (SCITX pin is a general-purpose output.) 0=Output value on SCITX is a 0 (logic low). 1=Output value on SCITX is a 1 (logic high).
1	RX_DATA_OUT	R/W	0h	Contains 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 DATA DIR = 1 (SCIRX pin is a general-purpose output.) 0=Output value on SCIRX is 0 (logic low). 1=Output value on SCIRX is 1 (logic high).
0	CLK_DATA_OUT	R/W	0h	Contains the data to be output on pin SCICLK if the following conditions are met: CLK FUNC = 0 (SCICLK pin is a general-purpose I/O.) CLK DATA DIR = 1 (SCICLK pin is a general-purpose output.) 0=Output value on SCICLK is a 0 (logic low). 1=Output value on SCICLK is a 1 (logic high).

15.6.20 SCIPIO4 Register (Offset = 4Ch) [Reset = 00000000h]

SCIPIO4 is shown in Table 15-24.

Return to the Summary Table.

SCI Pin I/O Control Register 4

Table 15-24. SCIPIO4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved

Table 15-24. SCIPIO4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	TX_DATA_SET	R/W	0h	Sets the data 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.)
1	RX_DATA_SET	R/W	0h	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.)
0	CLK_DATA_SET	R/W	0h	Sets the data to be output on pin SCICLK if the following conditions are met: CLK FUNC = 0 (SCICLK pin is a general-purpose I/O.) CLK DIR = 1 (SCICLK pin is a general-purpose output.)

15.6.21 SCIPIO5 Register (Offset = 50h) [Reset = 00000000h]

SCIPIO5 is shown in Table 15-25.

Return to the Summary Table.

SCI Pin I/O Control Register 5

Table 15-25. SCIPIO5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DATA_CLR	R/W	0h	Clears the data 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.)
1	RX_DATA_CLR	R/W	0h	Clears the data to be output on pin SCIRX if the following conditions are met: RX FUNC = 0 (SCITX pin is a general-purpose I/O.) RX DIR = 1 (SCITX pin is a general-purpose output.)
0	CLK_DATA_CLR	R/W	0h	Clears the data 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.)

15.6.22 SCIPIO6 Register (Offset = 54h) [Reset = 00000000h]

SCIPIO6 is shown in Table 15-26.

Return to the Summary Table.

SCI Pin I/O Control Register 6

Table 15-26. SCIPIO6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_PDR	R/W	0h	TX Open Drain Enable Enables open-drain capability in the output pin SCITX if the following conditions are met: TX DATA DIR = 1 (SCITX pin is a general-purpose output.) TX DOUT = 1
1	RX_PDR	R/W	0h	RX Open Drain Enable Enables open-drain capability in the output pin SCIRX if the following conditions are met: RX DATA DIR = 1 (SCIRX pin is a general-purpose output.) RX DOUT = 1
0	CLK_PDR	R/W	0h	CLK Open Drain Enable Enables open-drain capability in the output pin SCICLK if the following conditions are met: CLK DATA DIR = 1 (SCICLK pin is a general-purpose output.) CLK DOUT = 1

15.6.23 SCIPIO7 Register (Offset = 58h) [Reset = 00000000h]

SCIPIO7 is shown in Table 15-27.

Return to the Summary Table.

SCI Pin I/O Control Register 7

Table 15-27. SCIPIO7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_PD	R/W	0h	TX pin Pull Control Disable Disables pull control capability in the output pin SCITX. 0=Pull Control on SCITX pin is enabled. 1=Pull Control on SCITX pin is disabled.
1	RX_PD	R/W	0h	RX pin Pull Control Disable Disables pull control capability in the output pin SCIRX. 0=Pull Control on SCIRX pin is enabled. 1=Pull Control on SCIRX pin is disabled.
0	CLK_PD	R/W	0h	CLK pin Pull Control Disable Disables pull control capability in the output pin SCICLK. 0=Pull Control on SCICLK pin is enabled. 1=Pull Control on SCICLK pin is disabled.

15.6.24 SCIPIO8 Register (Offset = 5Ch) [Reset = 00000000h]

SCIPIO8 is shown in Table 15-28.

Return to the Summary Table.

SCI Pin I/O Control Register 8

Table 15-28. SCIPIO8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_PSL	R/W	0h	TX pin Pull Select Selects pull type in the output pin SCITX. 0=Pull-Down is on SCITX pin. 1=Pull-Up is on SCITX pin.
1	RX_PSL	R/W	0h	RX pin Pull Select Selects pull type in the output pin SCIRX. 0=Pull-Down is on SCIRX pin. 1=Pull-Up is on SCIRX pin.
0	CLK_PSL	R/W	0h	CLK pin Pull Select Selects pull type in the output pin SCICLK. 0=Pull-Down is on SCICLK pin. 1=Pull-Up is on SCICLK pin.

15.6.25 RESERVED2 Register (Offset = 60h) [Reset = 00000000h]

RESERVED2 is shown in Table 15-29.

Return to the Summary Table.

Reserved

Table 15-29. RESERVED2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.26 RESERVED3 Register (Offset = 64h) [Reset = 00000000h]

RESERVED3 is shown in Table 15-30.

Return to the Summary Table.

Reserved

Table 15-30. RESERVED3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.27 RESERVED4 Register (Offset = 68h) [Reset = 00000000h]

RESERVED4 is shown in Table 15-31.

Return to the Summary Table.

Reserved

Table 15-31. RESERVED4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.28 RESERVED5 Register (Offset = 6Ch) [Reset = 00000000h]

RESERVED5 is shown in Table 15-32.

Return to the Summary Table.

Reserved

Table 15-32. RESERVED5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.29 RESERVED6 Register (Offset = 70h) [Reset = 00000000h]

RESERVED6 is shown in Table 15-33.

Return to the Summary Table.

Reserved

Table 15-33. RESERVED6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.30 RESERVED7 Register (Offset = 74h) [Reset = 00000000h]

RESERVED7 is shown in Table 15-34.

Return to the Summary Table.

Reserved

Table 15-34. RESERVED7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.31 RESERVED8 Register (Offset = 78h) [Reset = 00000000h]

RESERVED8 is shown in Table 15-35.

Return to the Summary Table.

Reserved

Table 15-35. RESERVED8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.32 RESERVED9 Register (Offset = 7Ch) [Reset = 00000000h]

RESERVED9 is shown in Table 15-36.

Return to the Summary Table.

Reserved

Table 15-36. RESERVED9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

15.6.33 SCIPIO9 Register (Offset = 80h) [Reset = 00000000h]

SCIPIO9 is shown in Table 15-37.

Return to the Summary Table.

SCI Pin I/O Control Register 9

Table 15-37. SCIPIO9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_SL	R/W	0h	This bit controls the slew rate for the SCITX pin. 0=The normal output buffer is used for SCITX pin 1=The output buffer with slew control is used for SCITX pin.
1	RX_SL	R/W	0h	This bit controls the slew rate for the SCIRX pin. 0=The normal output buffer is used for SCIRX pin 1=The output buffer with slew control is used for SCIRX pin
0	CLK_SL	R/W	0h	This bit controls the slew rate for the SCICLK pin. 0=The normal output buffer is used for SCICLK pin 1=The output buffer with slew control is used for SCICLK pin

15.6.34 SCIIODCTRL Register (Offset = 90h) [Reset = 00000000h]

SCIIODCTRL is shown in Table 15-38.

Return to the Summary Table.

SCI IO DFT Control

Table 15-38. SCIIODCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	FEN	R/W	0h	Frame Error Enable. User and Privileged Mode Reads and Writes: 1 = This bit is used to create a Frame Error. The stop bit received is ANDed with ’0’ and passed to the stop bit check circuitry. 0 = No effect.
25	PEN	R/W	0h	Parity Error Enable. User and Privileged Mode Reads and Writes: 1 = This bit is used to create a Parity Error. The parity bit received is toggled so that a parity error occurs. 0 = No effect

Table 15-38. SCIIODCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	BRKDT_ENA	R/W	0h	Break Detect Error Enable. User and Privileged Mode Reads and Writes: 1 = This bit is used to create BRKDT Error. 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 TBITS so that a BRKDT error occurs. 0 = No effect.
23-21	RESERVED	R	0h	Reserved
20-19	PIN_SAMPLE_MASK	R/W	0h	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. PIN SAMPLE MASK: 00 -- No Mask, 01 -- Invert the TX Pin value at 7th SCLK, 10 -- Invert the TX Pin value at 8th SCLK, 11 -- Invert the TX Pin value at 9th SCLK.
18-16	TX_SHIFT	R/W	0h	These bits define the delay by which the value on TX pin is delayed so that the value on RX Pin is asynchronous. (Not applicable to Start Bit) TX SHIFT: 000 -- No Delay, 001 -- Delay by 1 SCLK, 010 -- Delay by 2 SCLKs, 011 -- Delay by 3 SCLKs, 100 -- Delay by 4 SCLKs, 101 -- Delay by 5 SCLKs, 110 -- Delay by 6 SCLKs, 111 -- No Delay.
15-12	RESERVED	R	0h	Reserved
11-8	IODFTENA	R/W	0h	These bits define the delay by which the value on TX pin is delayed so that the value on RX Pin is asynchronous. (Not applicable to Start Bit) TX SHIFT: 000 -- No Delay, 001 -- Delay by 1 SCLK, 010 -- Delay by 2 SCLKs, 011 -- Delay by 3 SCLKs, 100 -- Delay by 4 SCLKs, 101 -- Delay by 5 SCLKs, 110 -- Delay by 6 SCLKs, 111 -- No Delay.
7-2	RESERVED	R	0h	Reserved
1	LBP_ENA	R/W	0h	Module loopback enable. user and privileged mode reads: Write only in privileged mode: write/ read : 1=Analog loopback is enabled in module I/O DFT mode(when IODFTENA = 1010) 0=Digital loopback is enabled.
0	RXP_ENA	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.

15.7 SCI GPIO Functionality

The following sections apply to all device pins that can be configured as functional or general-purpose I/O pins.

15.7.1 GPIO Functionality

Figure 15-8 illustrates the GPIO functionality.

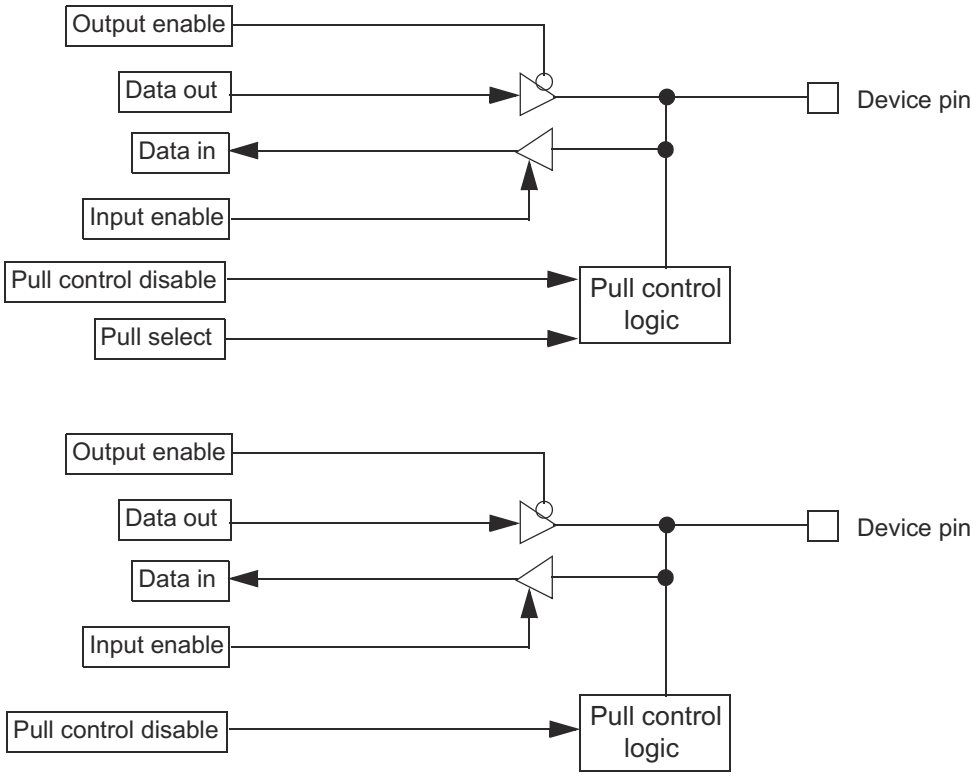


Figure 15-8. GPIO Functionality

15.7.2 Under Reset

The following apply if a device is under reset:

- Pull control. The reset pull control on the pins is enabled.
- Input buffer. The input buffer is enabled.
- Output buffer. The output buffer is disabled.

15.7.3 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 (). In this case, if the PSL (pull select) bit in the SCIPIO8 register () 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 ().

- 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;) AND the open-drain feature is not enabled in the SCIPIO6 register ().

15.7.4 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;).

15.7.5 Summary

The behavior of the input buffer, output buffer, and the pull control is summarized in Table 15-39.

Table 15-39. 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	Enabled	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

ADVANCE INFORMATION

16 Inter-Integrated Circuit (I2C) Module

This chapter describes the inter-integrated circuit (I2C or I^{2}C) module. The I2C is a multi-Target communication module providing an interface between the Texas Instruments (TI) microcontroller and devices compliant with Philips Semiconductor I^{2}C-bus specification version 2.1 and connected by an I2C-bus. This module will support any Controller or Target I2C compatible device.

16.1 Overview

The I2C has the following features:

- Compliance to the Philips (now NXP Semiconductors) I^{2}C bus specification, v2.1 (*The I 2 C Specification,* Philips document number 9398 393 40011)
  - Bit/Byte format transfer
  - 7-bit and 10-bit device addressing modes
  - General call
  - START byte
  - Multi-controller transmitter/target receiver mode
  - Multi-controller receiver/target transmitter mode
  - Combined controller transmit/receive and receive/transmit mode
  - Transfer rates of 10 kbps up to 400 kbps (Fast mode transfer rate)
- Free data format
- Two DMA events (transmit and receive)
- DMA event enable/disable capability
- Seven interrupts that can be used by the CPU
- Operates with VBUS frequency from 6.7 MHz up
- Operates with module frequency between 6.7 MHz and 13.3 MHz
- 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 (only supports up to 400 kbps (Fast mode))
- C-bus compatibility mode
- The combined format in 10-bit address mode (the I2C sends the target address second byte every time it sends the target address first byte)

16.1.1 Introduction to the I2C Module

The I2C module supports any target or controller I2C-compatible device. Figure 16-1 shows an example of multiple I2C serial ports connected for a two-way transfer from one device to another device.

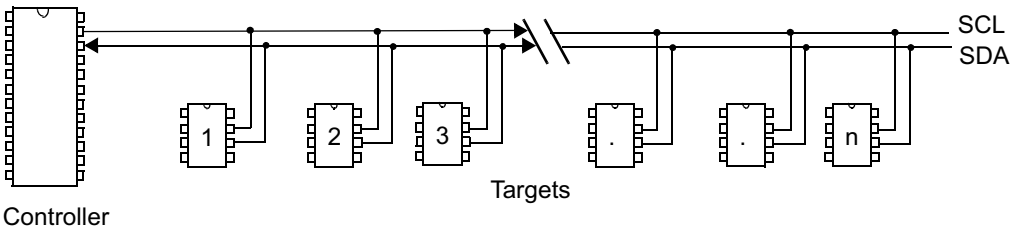


Figure 16-1. Multiple I2C Modules Connection Diagram

16.1.2 Functional Overview

The I2C module is a serial bus that supports multiple Controller devices. In multi Controller 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 Controller or a Target when performing data transfers.

Note

A Controller 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 Controller is considered the Target.

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 16-2. 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 Controller 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 Controller) and another Controller
- 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 controllers 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 16-2, 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

ADVANCE INFORMATION

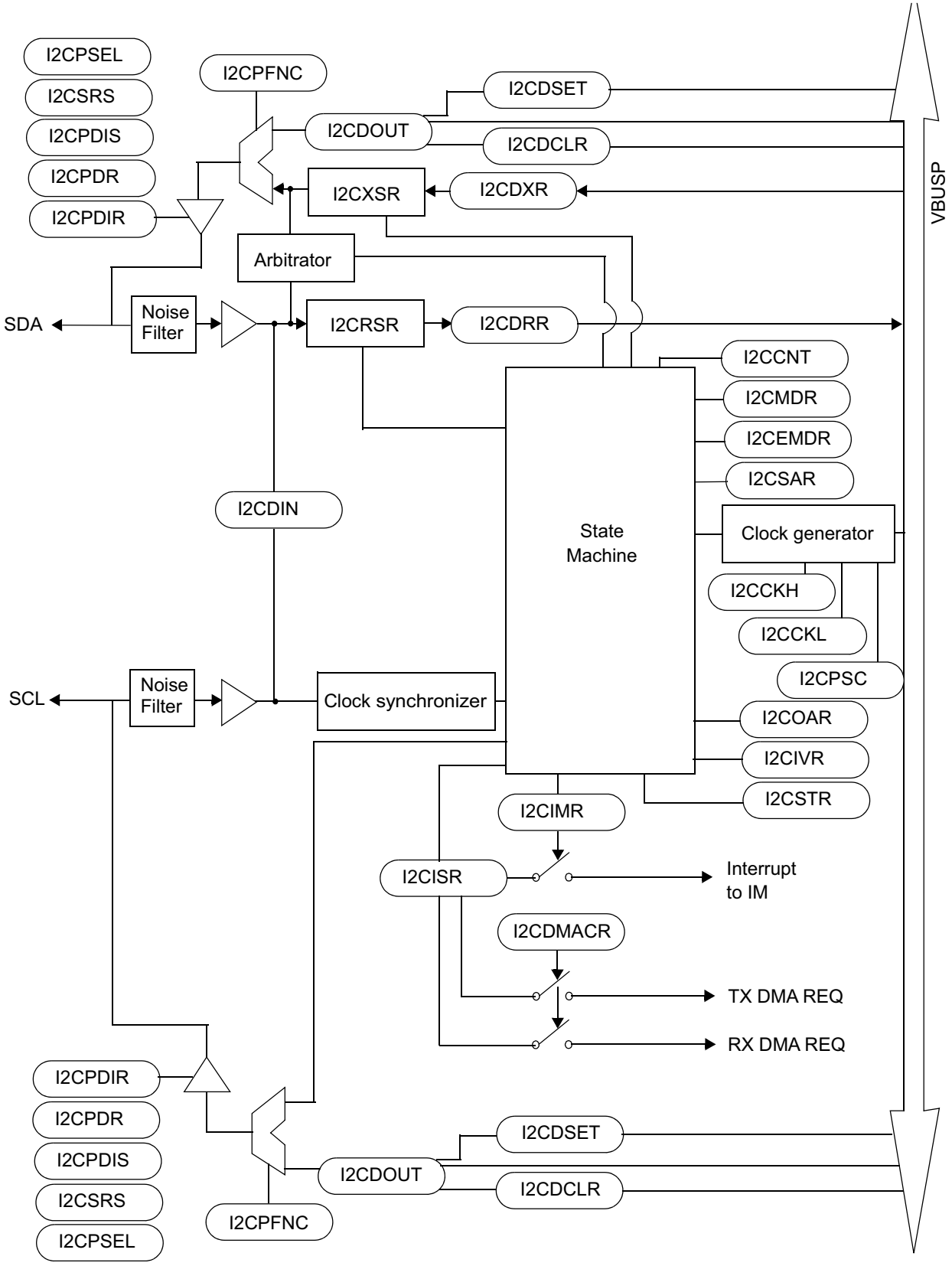


Figure 16-2. Simple I2C Block Diagram

16.1.3 Clock Generation

As shown in Figure 16-3, the I2C module uses the input clock generated from the device clock generator to generate the module clock and Controller 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 Controller clock.

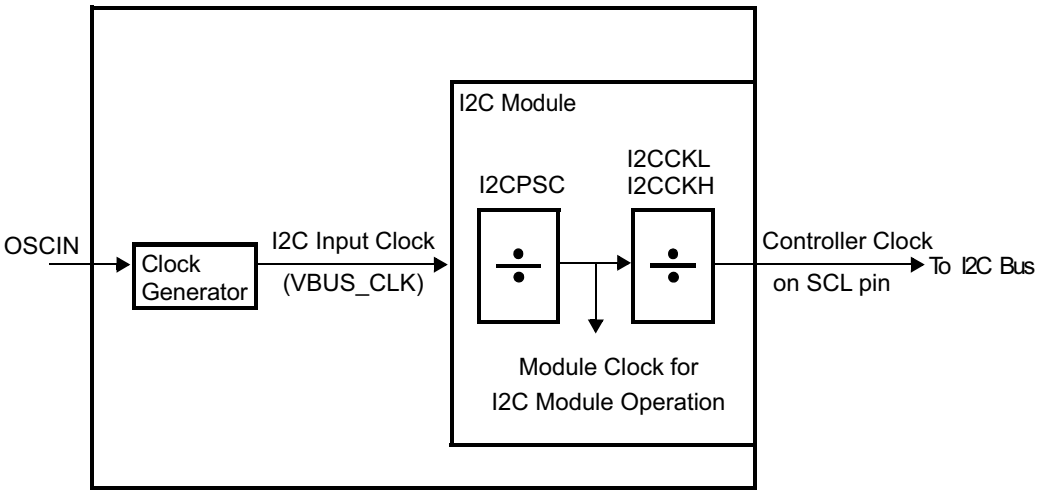


Figure 16-3. Clocking Diagram for the I2C Module

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:

$$ModuleClockFrequency = \frac{I2CInputClockFrequency}{(I2CPSC + 1)} \quad (6)$$

The module clock frequency must be between 6.7MHz and 13.3MHz. 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 Controller clock appears on the SCL pin when the I2C module is configured to be a Controller on the I2C bus. This clock controls the timing of the communication between the I2C module and a secondary. As shown in Figure 16-3, a second clock divider in the I2C module divides down the module clock to produce the Controller 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:

$$ControllerClockFrequency = \frac{ModuleClockFrequency}{(I2CCKL + d) + (I2CCKH + d)} \quad (7)$$

$$ControllerClockFrequency = \frac{I2CInputClockFrequency}{(I2CPSC + 1)((I2CCKL + d) + (I2CCKH + d))} \quad (8)$$

where *d* depends on the value of I2CPSC:

I2CPSC	d
0	7
1	6

I2CPSC	d
Greater than 1	5

Note

The Controller 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.

16.2 I2C Module Operation

The following section discusses how the I2C module operates.

16.2.1 Input and Output Voltage Levels

One clock pulse is generated by the Controller 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.

16.2.2 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 $\overline{IRS}$ bit in the I2C mode register (I2CMDR). When the global peripheral reset is removed, the $\overline{IRS}$ bit is cleared to 0, keeping the I2C module in the reset state.

16.2.3 I2C Module Data Validity

The data on the SDA must be stable during the high period of the clock. See Figure 16-4. The high and low state of the data line, the SDA, can only change when the clock signal on the serial clock line (SCL) is low.

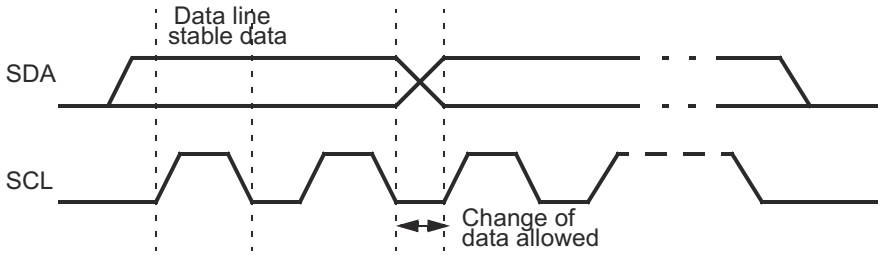


Figure 16-4. Bit Transfer on the I2C Bus

ADVANCE INFORMATION

16.2.4 I2C Module Start and Stop Conditions

START and STOP conditions are generated by a primary I2C module.

- The START condition is defined as a high-to-low transition on the SDA line while SCL is high. A primary 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 primary 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.

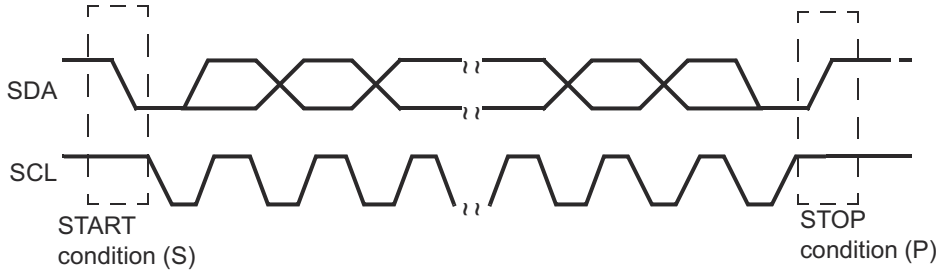


Figure 16-5. I2C Module START and STOP Conditions

For the I2C module to start a data transfer with a START condition, the primary 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.

16.2.5 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 16-6). 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.

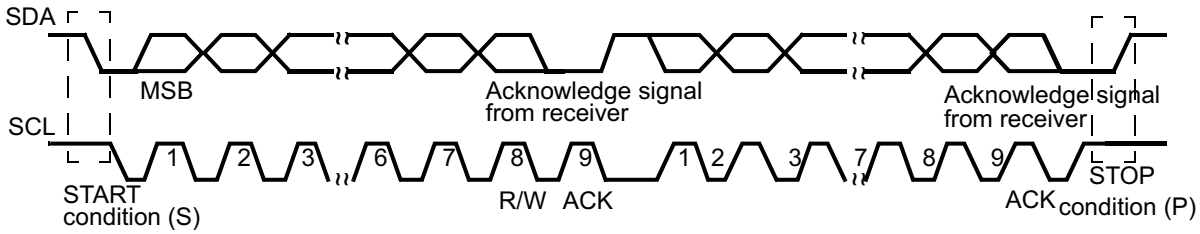


Figure 16-6. I2C Module Data Transfer

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 Controller writes (transmits) data to a selected Target device; when the R/ $\overline{W}$ bit is 1, the Controller reads (receives) data from the Target device. In acknowledge mode, an extra bit dedicated for the acknowledgment (ACK) bit is inserted after each message.

The I2C module supports the following formats:

- 7-bit addressing format (Figure 16-7)
- 10-bit addressing format (Figure 16-8)
- 7-bit/10-bit addressing format with repeated START condition (Figure 16-9)
- Free-data format (Figure 16-10)

ADVANCE INFORMATION

16.2.5.1 7-Bit Addressing Format

In the 7-bit addressing format (Figure 16-7), the first byte after the START condition consists of a 7-bit secondary 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 primary writes (transmits) data to the addressed secondary.
- R/ $\overline{W}$ = 1: The primary reads (receives) data from the secondary.

An extra clock cycle dedicated for acknowledgement (ACK) is inserted after each byte. If the ACK is inserted by the secondary after the first byte from the primary, it is followed by n bits of data from the transmitter (primary or secondary, 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).

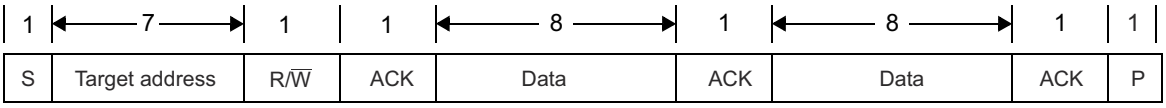


Figure 16-7. I2C Module 7-Bit Addressing Format

16.2.5.2 10-Bit Addressing Format

The 10-bit addressing format is similar to the 7-bit addressing format, but the primary sends the secondary address in two separate byte transfers. In the 10-bit addressing format (Figure 16-8), the first byte is 11110b, the two MSBs of the 10-bit secondary 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 secondary address. The secondary must send an acknowledgment after each of the two byte transfers. Once the primary has written the second byte to the secondary, the primary 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 16-8. I2C Module 10-bit Addressing Format

16.2.5.3 Using the Repeated START Condition

At the end of each byte, the primary can drive another START condition (Figure 16-9). Using this capability, a primary 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.

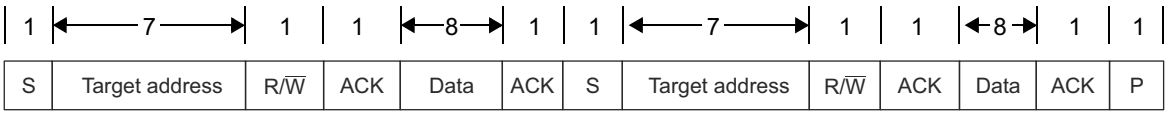


Figure 16-9. I2C Module 7-Bit Addressing Format with Repeated START

16.2.5.4 Free Data Format

In this format (Figure 16-10), 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.

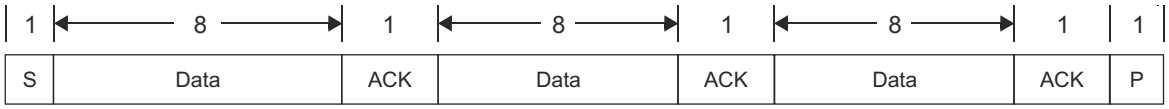


Figure 16-10. I2C Module in Free Data Format

16.2.6 NACK Bit Generation

When the I2C module is a receiver (Controller or Target), 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 16-1 summarizes the various ways a NACK can be generated.

Table 16-1. Ways to Generate a NACK Bit

I2C Module Condition	Basic NACK Bit Generation Options	Additional Option
Target 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.
Controller 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.
Controller 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 Target 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.

16.3 I2C Operation Modes

16.3.1 Controller Transmitter Mode

All primaries begin in this mode. The I2C module is a Controller and transmits control information and data to a Target. In this mode, data assembled in any of the addressing formats shown in Figure 16-7, Figure 16-8, or Figure 16-9 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 ($\overline{\text{XSMT}}$ = 0) after a byte has been transmitted.

Note

If the I2C is configured for two simultaneous Controller transmissions, wait until the MST and BB have been reset before performing the second Controller 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 Controller 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 Controller, thus failing to occupy the bus.

16.3.2 Controller Receiver Mode

In this mode, the I2C module is a Controller and receives data from a Target. This mode can only be entered from the Controller transmitter mode (the I2C module must first transmit a command to the Target). In any of the addressing formats shown in Figure 16-7, Figure 16-8, or Figure 16-9, the Controller receiver mode is entered after the Target address byte and the R/ $\overline{\text{W}}$ bit have been transmitted (if the R/ $\overline{\text{W}}$ 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 Controller-receiver signals the end of data to the Target-transmitter by not generating an acknowledge on the last byte that was clocked out of the Target. The Target-transmitter then releases the data line allowing the Controller-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 Controller before the transmission/reception starts, the Controller must use the repeat mode 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 Controller must generate the stop condition (STP =1) after reading the (message size - 1)th data.

16.3.3 Target Transmitter Mode

In this mode, the I2C module is a Target and transmits data to a Controller. This mode can only be entered from the Target receiver mode (The I2C module must first receive a command from the Controller). In any of the addressing formats shown in Figure 16-7, Figure 16-8, or Figure 16-9, the Target transmitter mode is entered if the Target address byte is the same as its own address and the R/ $\overline{\text{W}}$ bit has been transmitted (if the R/ $\overline{\text{W}}$ bit is set to 1). The Target transmitter shifts the serial data out on the SDA pin with the clock pulses that are generated by the Controller device. The Target device does not generate the clock, but it can hold the SCL pin low when intervention of the device is required ($\overline{\text{XSMT}}$ = 0) after a byte has been transmitted.

16.3.4 Target Receiver Mode

In this mode, the I2C module is a Target and receives data from a Controller. All Target begin in this mode. Serial data bits received on the SDA pin are shifted in with the clock pulses that are generated by the Controller device. The Target 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.

16.3.5 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 break point 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 Controller 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 Target, it stops when the transmission/reception completes.

16.3.6 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 Controller transmitter to ignore a NACK from a Target device that is not capable of generating a proper ACK signal.

16.4 I2C Module Integrity

The following section discusses how the I2C module maintains priorities and order among signals and commands.

16.4.1 Arbitration

If two or more Controller transmitters simultaneously start a transmission on the same bus, an arbitration procedure is invoked. Figure 16-11 illustrates the arbitration procedure between two devices. The arbitration procedure uses the data presented on the SDA bus by the competing transmitters. The first Controller transmitter that generates a high is overruled by the other Controller that generates a low. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. The Controller transmitter that loses the arbitration switches to the Target receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration-lost interrupt. The data transmitted by the other Controller module is salvaged, and the I2C continues to receive data from the Controller 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 Controller 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

Secondarys are not involved in the arbitration procedure.

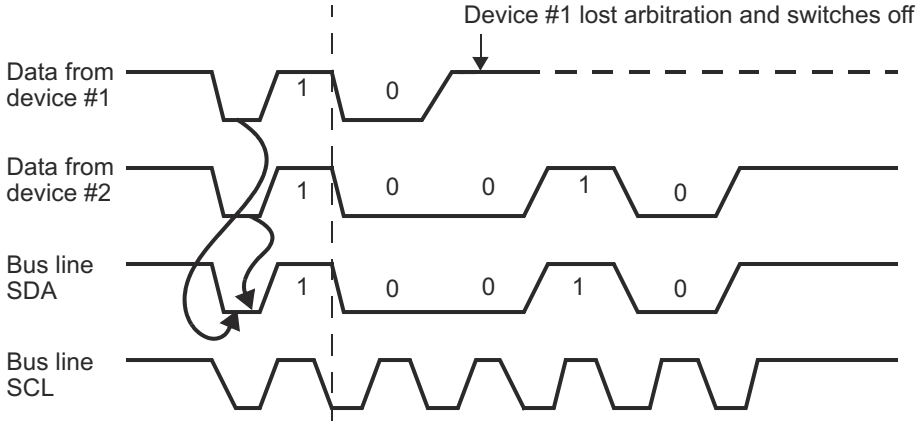


Figure 16-11. Arbitration Procedure Between Two Controller Transmitters

16.4.2 I2C Clock Generation and Synchronization

Under normal conditions only one Controller device generates the clock signal; the SCL. During the arbitration procedure, however, there are two or more Controller devices and the clock must be synchronized so that the data output can be compared. Figure 16-12 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 Target slows down a fast Controller 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 I^2C bus specification, v2.1 (*The I 2 C 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.

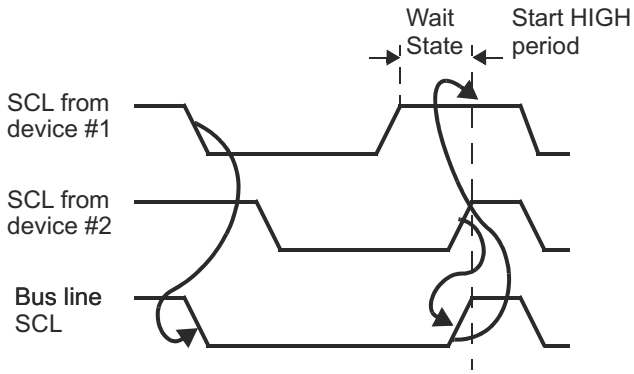


Figure 16-12. Synchronization of Two I2C Clock Generators During Arbitration

16.4.3 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.

16.4.4 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.

16.5 Operational Information

The following section provides specific information about how the I2C module operates.

16.5.1 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).

16.5.1.1 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 16-2.

Table 16-2. 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 Controller-transmitter
NACK	No-acknowledge interrupt	Generated when the Controller 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-Target interrupt	Generated when the I2C has recognized its own Target address or an address of all zeroes.

16.5.2 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 Controller transmitter mode. This ensures that the DMA gets an event even if no Target returns an ACK to the Target 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, Controller 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 Target ACK being received.

16.5.3 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.

16.5.4 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.

16.5.5 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 enabled. Please see the device-specific data sheet for the current supplied by the pull up/pull down.

16.5.6 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.

16.6 APP_I2C Registers

Table 16-3 lists the memory-mapped registers for the APP_I2C registers. All register offset addresses not listed in Table 16-3 should be considered as reserved locations and the register contents should not be modified.

Table 16-3. APP_I2C Registers

Offset	Acronym	Register Name	Section
0h	ICOAR	ICOAR	Go
4h	ICIMR	ICIMR	Go
8h	ICSTR	ICSTR	Go
Ch	ICCLKL	ICCLKL	Go
10h	ICCLKH	ICCLKH	Go
14h	ICCNT	ICCNT	Go
18h	ICDRR	ICDRR	Go
1Ch	ICSAR	ICSAR	Go
20h	ICDXR	ICDXR	Go
24h	ICMDR	ICMDR	Go
28h	ICIVR	ICIVR	Go
2Ch	ICEMDR	ICEMDR	Go
30h	ICPSC	ICPSC	Go
34h	ICPID1	ICPID1	Go
38h	ICPID2	ICPID2	Go
3Ch	ICDMAC	ICDMAC	Go
40h	I2C_RESERVED1	I2C_RESERVED1	Go
44h	I2C_RESERVED2	I2C_RESERVED2	Go
48h	ICPFUNC	ICPFUNC	Go
4Ch	ICPDIR	ICPDIR	Go
50h	ICPDIN	ICPDIN	Go
54h	ICPDOUT	ICPDOUT	Go
58h	ICPDSET	ICPDSET	Go
5Ch	ICPDCLR	ICPDCLR	Go
60h	ICPDRV	ICPDRV	Go

Complex bit access types are encoded to fit into small table cells. Table 16-4 shows the codes that are used for access types in this section.

Table 16-4. APP_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

16.6.1 ICOAR Register (Offset = 0h) [Reset = 00000000h]

ICOAR is shown in Table 16-5.

Return to the Summary Table.

I2C Own Address register

Table 16-5. ICOAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU	R	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.

16.6.2 ICIMR Register (Offset = 4h) [Reset = 00000000h]

ICIMR is shown in Table 16-6.

Return to the Summary Table.

I2C Interrupt Mask/Status register

Table 16-6. ICIMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	NU	R	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.

16.6.3 ICSTR Register (Offset = 8h) [Reset = 00000410h]

ICSTR is shown in Table 16-7.

Return to the Summary Table.

I2C Interrupt Status register

Table 16-7. ICSTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	NU2	R	0h	Reserved

Table 16-7. ICSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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	1h	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.
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)

Table 16-7. ICSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	NU1	R	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	1h	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.

16.6.4 ICCLKL Register (Offset = Ch) [Reset = 00000000h]

ICCLKL is shown in Table 16-8.

Return to the Summary Table.

I2C Clock Divider Low register

Table 16-8. ICCLKL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	Reserved

Table 16-8. ICCLKL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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).

16.6.5 ICCLKH Register (Offset = 10h) [Reset = 00000000h]

ICCLKH is shown in Table 16-9.

Return to the Summary Table.

I2C Clock Divider High register

Table 16-9. ICCLKH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	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).

16.6.6 ICCNT Register (Offset = 14h) [Reset = 00000000h]

ICCNT is shown in Table 16-10.

Return to the Summary Table.

I2C Data Count register

Table 16-10. ICCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	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.

16.6.7 ICDRR Register (Offset = 18h) [Reset = 00000000h]

ICDRR is shown in Table 16-11.

Return to the Summary Table.

I2C Data Receive register

Table 16-11. ICDRR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Reserved
7-0	D7_D0	R/W	0h	Receive data

16.6.8 ICSAR Register (Offset = 1Ch) [Reset = 000003FFh]

ICSAR is shown in Table 16-12.

Return to the Summary Table.

I2C Slave Address register

Table 16-12. ICSAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU	R	0h	Reserved
9-0	A9_A0	R/W	3FFh	Slave address. Use in both 7- and 10-bit address mode.

16.6.9 ICDXR Register (Offset = 20h) [Reset = 00000000h]

ICDXR is shown in Table 16-13.

Return to the Summary Table.

I2C Data Transmit register

Table 16-13. ICDXR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Reserved
7-0	D7_D0	R/W	0h	Transmit data

16.6.10 ICMDR Register (Offset = 24h) [Reset = 00000000h]

ICMDR is shown in Table 16-14.

Return to the Summary Table.

I2C Mode register

Table 16-14. ICMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU2	R	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.

Table 16-14. ICMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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	0h	Reserved for IDLEEN (IDLE Enable on 5509). - (RW)
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" _____

ADVANCE INFORMATION

Table 16-14. ICMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.
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 _____

ADVANCE INFORMATION

Table 16-14. ICMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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_____

16.6.11 ICIVR Register (Offset = 28h) [Reset = 00000000h]

ICIVR is shown in Table 16-15.

Return to the Summary Table.

I2C Interrupt Vector register

Table 16-15. ICIVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU2	R	0h	Reserved.
11-8	TESTMD	R/W	0h	Reserved for internal testing.
7-3	NU1	R	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)

16.6.12 ICEMDR Register (Offset = 2Ch) [Reset = 00000001h]

ICEMDR is shown in Table 16-16.

Return to the Summary Table.

I2C Extended Mode register

ADVANCE INFORMATION

Table 16-16. ICEMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	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	1h	Backward Compatibility Mode. This bit affects the I2C interrupt behavior. Refer to appendix A for details.

16.6.13 ICPSC Register (Offset = 30h) [Reset = 00000000h]

ICPSC is shown in Table 16-17.

Return to the Summary Table.

I2C Prescaler register

Table 16-17. ICPSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	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_.

16.6.14 ICPID1 Register (Offset = 34h) [Reset = 00000146h]

ICPID1 is shown in Table 16-18.

Return to the Summary Table.

I2C Peripheral ID register 1

Table 16-18. ICPID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	Reserved.
15-8	CLASS	R/W	1h	Identifies the class of peripheral. This value should be 0x01 - (RW)
7-0	REVISION	R/W	46h	Identifies the revision level of the I2C. This value should be incremented each time the design is revised. - (RW)

16.6.15 ICPID2 Register (Offset = 38h) [Reset = 00000005h]

ICPID2 is shown in Table 16-19.

Return to the Summary Table.

I2C Peripheral ID register 2

Table 16-19. ICPID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	NU	R	0h	Reserved.

Table 16-19. ICPID2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	TYPE	R/W	5h	Identifies the type of peripheral. This value should be 0x05 - (RW)

16.6.16 ICDMAC Register (Offset = 3Ch) [Reset = 00000003h]

ICDMAC is shown in Table 16-20.

Return to the Summary Table.

I2C DMA Control Register

Table 16-20. ICDMAC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	0h	Reserved. - (RW)
1	TXDMAEN	R/W	1h	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	1h	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)

16.6.17 I2C_RESERVED1 Register (Offset = 40h) [Reset = 00000000h]

I2C_RESERVED1 is shown in Table 16-21.

Return to the Summary Table.

Reserved

Table 16-21. I2C_RESERVED1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU	R	0h	Reserved.

16.6.18 I2C_RESERVED2 Register (Offset = 44h) [Reset = 00000000h]

I2C_RESERVED2 is shown in Table 16-22.

Return to the Summary Table.

Reserved

Table 16-22. I2C_RESERVED2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU	R	0h	Reserved.

16.6.19 ICPFUNC Register (Offset = 48h) [Reset = 00000000h]

ICPFUNC is shown in Table 16-23.

Return to the Summary Table.

I2C Pin Function register

Table 16-23. ICPFUNC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	NU	R	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.

16.6.20 ICPDIR Register (Offset = 4Ch) [Reset = 00000000h]

ICPDIR is shown in Table 16-24.

Return to the Summary Table.

I2C Pin Direction register

Table 16-24. ICPDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	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

16.6.21 ICPDIN Register (Offset = 50h) [Reset = 00000000h]

ICPDIN is shown in Table 16-25.

Return to the Summary Table.

I2C Pin Data In register

Table 16-25. ICPDIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	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)

Table 16-25. ICPDIN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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)

16.6.22 ICPDOUT Register (Offset = 54h) [Reset = 00000000h]

ICPDOUT is shown in Table 16-26.

Return to the Summary Table.

I2C Pin Data Out register

Table 16-26. ICPDOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	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.

16.6.23 ICPDSET Register (Offset = 58h) [Reset = 00000000h]

ICPDSET is shown in Table 16-27.

Return to the Summary Table.

I2C Pin Data Set register

Table 16-27. ICPDSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	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.

16.6.24 ICPDCLR Register (Offset = 5Ch) [Reset = 00000000h]

ICPDCLR is shown in Table 16-28.

Return to the Summary Table.

I2C Pin Data Clear register

Table 16-28. ICPDCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	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.

16.6.25 ICPDRV Register (Offset = 60h) [Reset = 00000000h]

ICPDRV is shown in Table 16-29.

Return to the Summary Table.

I2C Pin Driver Mode Register

Table 16-29. ICPDRV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	NU	R	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.

16.7 Sample Waveforms

Figure 16-13 provides waveforms to illustrate the difference between normal operation and backward compatibility mode.

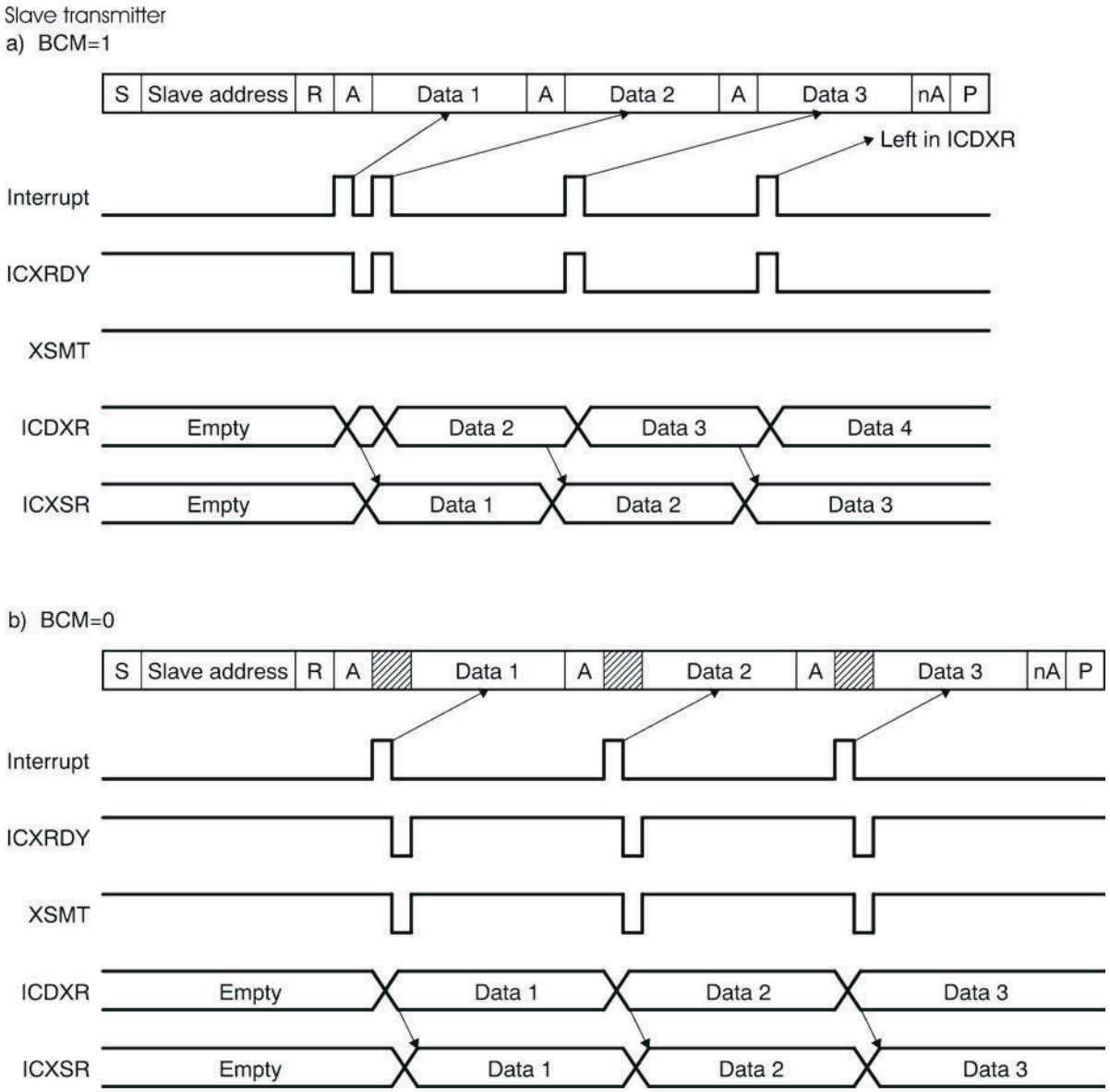


Figure 16-13. Difference between Normal Operation and Backward Compatibility Mode

ADVANCE INFORMATION

17 General-Purpose Input/Output (GIO) Module

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.

Note

The "GIO" module is also known as the "GPIO" module in other TI MCU and MPU devices. The two terms are used interchangeably and represent the general use I/O module of the device.

17.1 Overview

The GIO module offers general-purpose input and output capability. It supports up to eight 8-bit ports for a total of up to 64 GIO terminals. Each of these 64 terminals 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 up to 32 of these GIO terminals. Refer to the device datasheet for identifying the number of GIO ports supported and the GIO terminals capable of generating an interrupt.

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 up to 32 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.

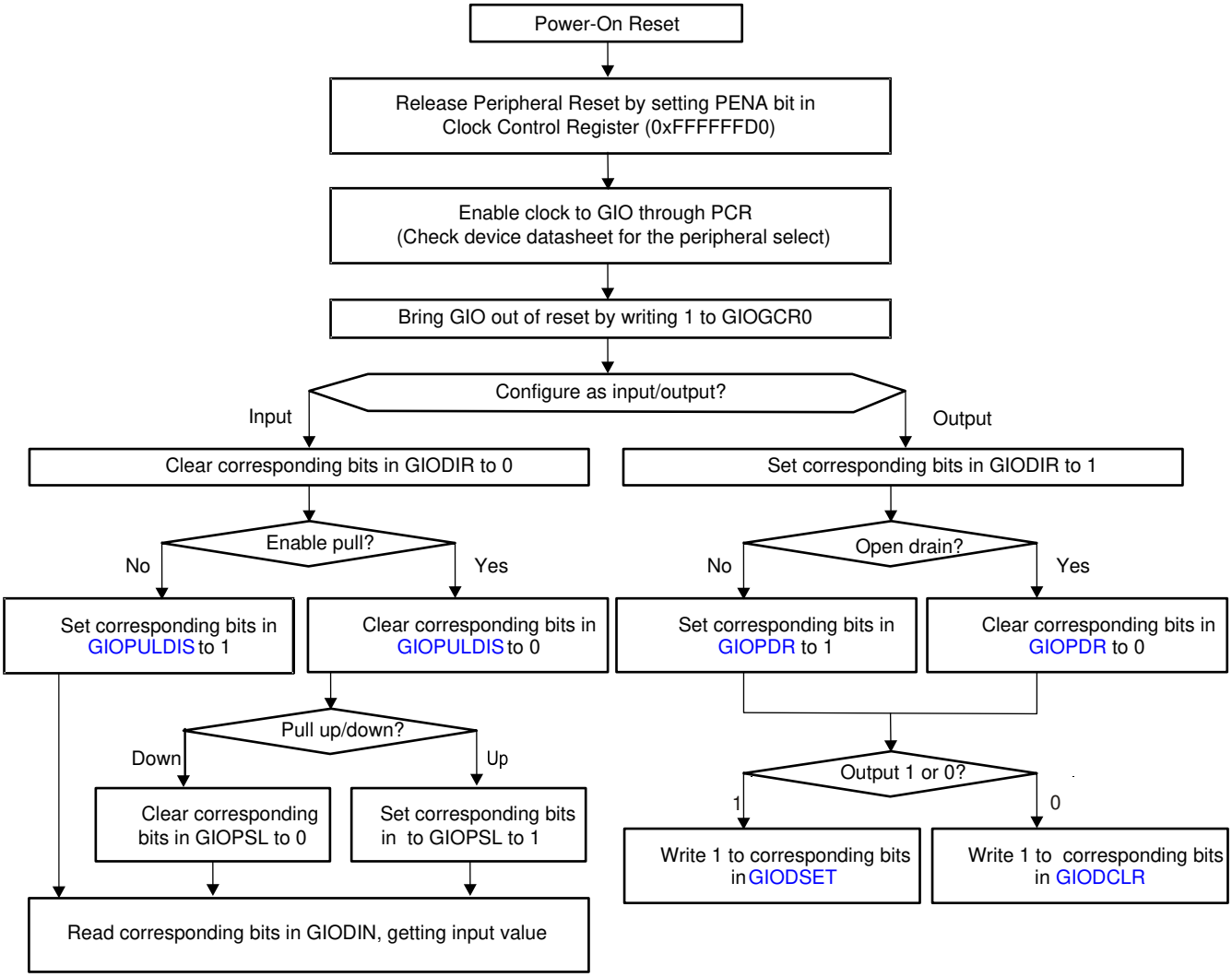


Figure 17-1. I/O Function Quick Start Flow Chart

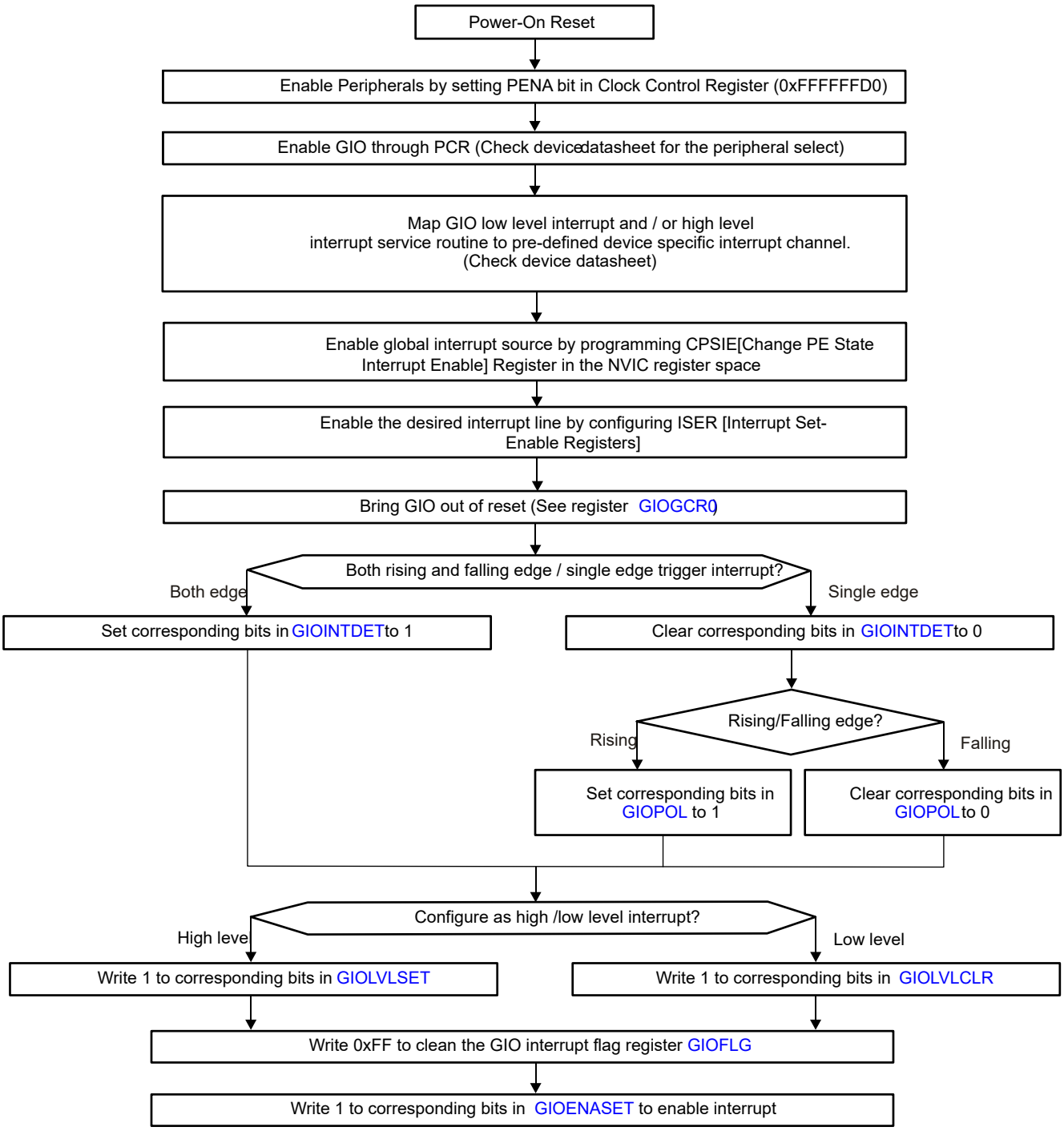


Figure 17-2. Interrupt Generation Function Quick Start Flow Chart

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.

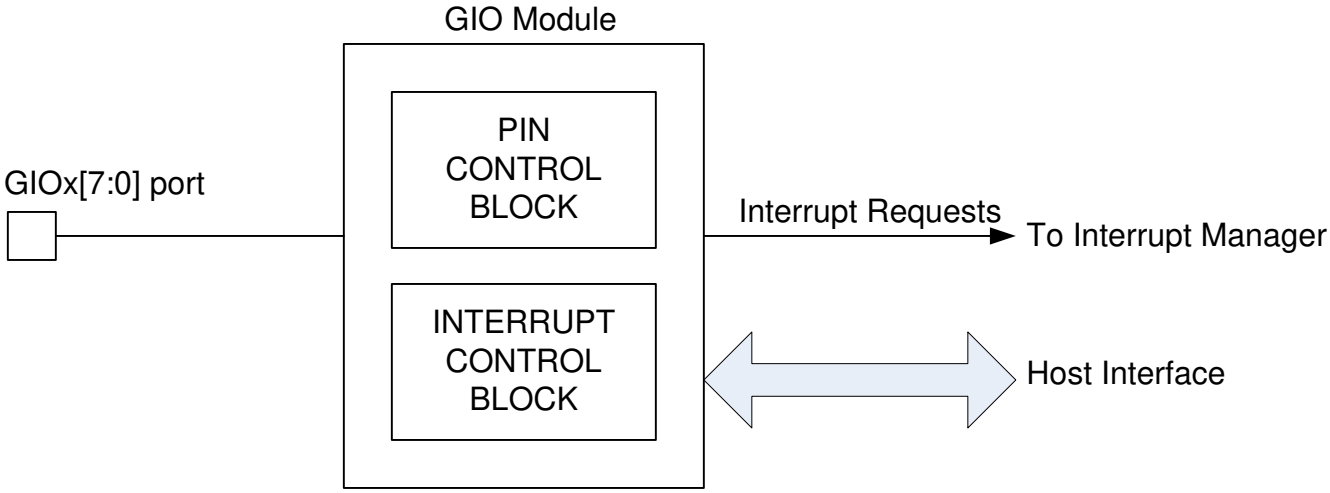


Figure 17-3. GIO Module Diagram

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).

ADVANCE INFORMATION

- 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 up to 32 terminals to be configured for generating an interrupt to the host processor through the Interrupt Manager (NVIC/IM). 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

The terminals on GIO ports A through D are all interrupt-capable and can be used to handle either general I/O functions or interrupt requests. Each interrupt request can be connected to the NVIC/IM at one of two different levels – High (or A) and Low (or B), depending on the NVIC/IM channel number. The NVIC/IM 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 NVIC/IM 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 NVIC/IM.

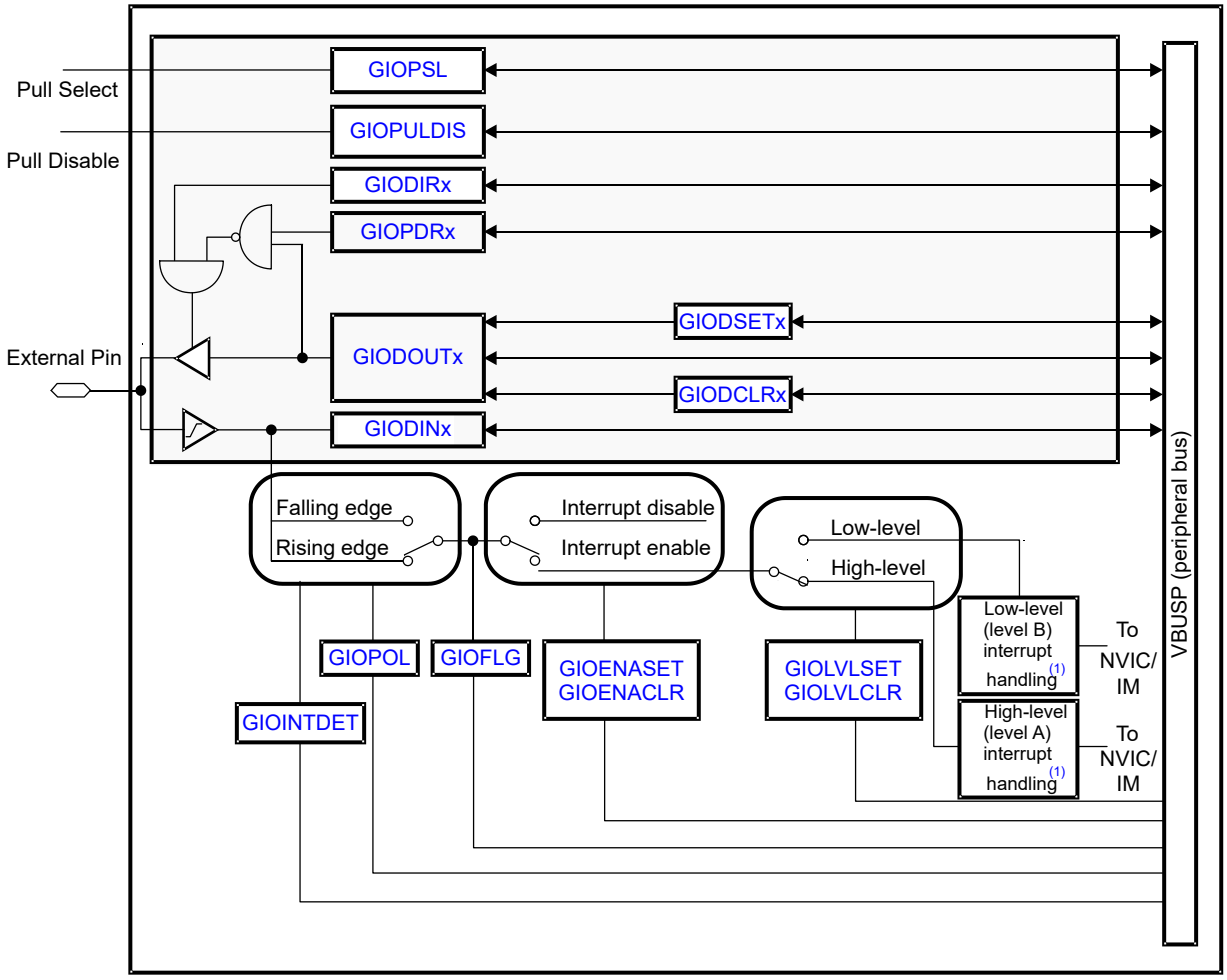
17.3.3 GIO Block Diagram

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

A. 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.

Figure 17-5.



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 NVIC/IM 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.

17.4.3 Interrupts

GIO generates aggregated interrupts for all inputs from PAD. Some of the interrupts to APPSS are directly taken from GPIO-PAD.

- APPSS_GIO_INT0/1
- GIO_INT1 : Interrupt trigger from GIO[0]
- GIO_INT2 : Interrupt trigger from GIO[1]

17.5 TOP_GIO Registers

Table 17-1 lists the memory-mapped registers for the TOP_GIO registers. All register offset addresses not listed in Table 17-1 should be considered as reserved locations and the register contents should not be modified.

Table 17-1. TOP_GIO Registers

Offset	Acronym	Register Name	Section
0h	GIOGCR	GIOGCR	Go
4h	GIOPWDN	GIOPWDN	Go
8h	GIOINTDET	GIOINTDET	Go
Ch	GIOPOL	GIOPOL	Go
10h	GIOENASET	GIOENASET	Go
14h	GIOENACLR	GIOENACLR	Go
18h	GIOLVLSET	GIOLVLSET	Go
1Ch	GIOLVLCLR	GIOLVLCLR	Go
20h	GIOFLG	GIOFLG	Go
24h	GIOOFFA	GIOOFFA	Go
28h	GIOOFFB	GIOOFFB	Go
2Ch	GIOEMUA	GIOEMUA	Go
30h	GIOEMUB	GIOEMUB	Go
34h	GIODIRA	GIODIRA	Go
38h	GIODINA	GIODINA	Go
3Ch	GIODOUTA	GIODOUTA	Go
40h	GIOSETA	GIOSETA	Go
44h	GIOCLRA	GIOCLRA	Go
48h	GIOPDRA	GIOPDRA	Go
4Ch	GIOPULDISA	GIOPULDISA	Go
50h	GIOPSLA	GIOPSLA	Go
54h	GIODIRB	GIODIRB	Go
58h	GIODINB	GIODINB	Go
5Ch	GIODOUTB	GIODOUTB	Go
60h	GIOSETB	GIOSETB	Go
64h	GIOCLRB	GIOCLRB	Go
68h	GIOPDRB	GIOPDRB	Go
6Ch	GIOPULDISB	GIOPULDISB	Go
70h	GIOPSLB	GIOPSLB	Go
74h	GIODIRC	GIODIRC	Go
78h	GIODINC	GIODINC	Go
7Ch	GIODOUTC	GIODOUTC	Go
80h	GIOSETC	GIOSETC	Go
84h	GIOCLRC	GIOCLRC	Go
88h	GIOPDRC	GIOPDRC	Go
8Ch	GIOPULDISC	GIOPULDISC	Go
90h	GIOPSLC	GIOPSLC	Go
94h	GIODIRD	GIODIRD	Go
98h	GIODIND	GIODIND	Go
9Ch	GIODOUTD	GIODOUTD	Go
A0h	GIOSETD	GIOSETD	Go
A4h	GIOCLRD	GIOCLRD	Go
A8h	GIOPDRD	GIOPDRD	Go

Table 17-1. TOP_GIO Registers (continued)

Offset	Acronym	Register Name	Section
ACh	GIOPULDISD	GIOPULDISD	Go
B0h	GIOPSLD	GIOPSLD	Go
B4h	GIODIRE	GIODIRE	Go
B8h	GIODINE	GIODINE	Go
BCh	GIODOUTE	GIODOUTE	Go
C0h	GIOSETE	GIOSETE	Go
C4h	GIOCLRE	GIOCLRE	Go
C8h	GIOPDRE	GIOPDRE	Go
CCh	GIOPULDISE	GIOPULDISE	Go
D0h	GIOPSLE	GIOPSLE	Go
D4h	GIODIRF	GIODIRF	Go
D8h	GIODINF	GIODINF	Go
DCh	GIODOUTF	GIODOUTF	Go
E0h	GIOSETF	GIOSETF	Go
E4h	GIOCLRF	GIOCLRF	Go
E8h	GIOPDRF	GIOPDRF	Go
ECh	GIOPULDISF	GIOPULDISF	Go
F0h	GIOPSLF	GIOPSLF	Go
F4h	GIODIRG	GIODIRG	Go
F8h	GIODING	GIODING	Go
FCh	GIODOUTG	GIODOUTG	Go
100h	GIOSETG	GIOSETG	Go
104h	GIOCLRG	GIOCLRG	Go
108h	GIOPDRG	GIOPDRG	Go
10Ch	GIOPULDISG	GIOPULDISG	Go
110h	GIOPSLG	GIOPSLG	Go
114h	GIODIRH	GIODIRH	Go
118h	GIODINH	GIODINH	Go
11Ch	GIODOUTH	GIODOUTH	Go
120h	GIOSETH	GIOSETH	Go
124h	GIOCLRH	GIOCLRH	Go
128h	GIOPDRH	GIOPDRH	Go
12Ch	GIOPULDISH	GIOPULDISH	Go
130h	GIOPSLH	GIOPSLH	Go
134h	GIOSRCA	GIOSRCA	Go
138h	GIOSRCB	GIOSRCB	Go
13Ch	GIOSRCC	GIOSRCC	Go
140h	GIOSRCD	GIOSRCD	Go
144h	GIOSRCE	GIOSRCE	Go
148h	GIOSRCF	GIOSRCF	Go
14Ch	GIOSRCG	GIOSRCG	Go
150h	GIOSRCH	GIOSRCH	Go

Complex bit access types are encoded to fit into small table cells. Table 17-2 shows the codes that are used for access types in this section.

Table 17-2. TOP_GIO 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

17.5.1 GIOGCR Register (Offset = 0h) [Reset = 00000000h]

GIOGCR is shown in Table 17-3.

Return to the Summary Table.

GIO reset

Table 17-3. 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 = 00000000h]

GIOPWDN is shown in Table 17-4.

Return to the Summary Table.

GIO power down mode register

Table 17-4. GIOPWDN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	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 = 00000000h]

GIOINTDET is shown in Table 17-5.

Return to the Summary Table.

Interrupt detection select for pins [0:1] GIO[7:0].

Table 17-5. 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 = 00000000h]

GIOPOL is shown in Table 17-6.

Return to the Summary Table.

Interrupt polarity select for pins [0:1] GIO[7:0].

Table 17-6. 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 = 00000000h]

GIOENASET is shown in Table 17-7.

Return to the Summary Table.

Interrupt enable for pins [0:1] GIO[7:0].

Table 17-7. 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 = 00000000h]

GIOENACLR is shown in Table 17-8.

Return to the Summary Table.

Interrupt enable for pins [0:1] GIO[7:0].

Table 17-8. 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 = 00000000h]

GIOLVLSET is shown in Table 17-9.

Return to the Summary Table.

GIO high priority interrupt for pins [0:1] GIO[7:0].

Table 17-9. 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 = 00000000h]

GIOLVLCLR is shown in Table 17-10.

Return to the Summary Table.

GIO low priority interrupt for pins [0:1] GIO[7:0].

Table 17-10. 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 = 00000000h]

GIOFLG is shown in Table 17-11.

Return to the Summary Table.

GIO flag for pins [0:1] GIO[7:0].

Table 17-11. 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 = 00000000h]

GIOOFFA is shown in Table 17-12.

Return to the Summary Table.

Index bits for currently pending high-priority interrupt Register A

Table 17-12. 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 = 00000000h]

GIOOFFB is shown in Table 17-13.

Return to the Summary Table.

Index bits for currently pending high-priority interrupt Register B

Table 17-13. 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 = 00000000h]

GIOEMUA is shown in Table 17-14.

Return to the Summary Table.

GIO emulation register A

Table 17-14. 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 = 00000000h]

GIOEMUB is shown in Table 17-15.

Return to the Summary Table.

GIO emulation register B

Table 17-15. 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 = 00000000h]

GIODIRA is shown in Table 17-16.

Return to the Summary Table.

GIO data direction of pins in Port A

Table 17-16. 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 = 00000000h]

GIODINA is shown in Table 17-17.

Return to the Summary Table.

GIO data input for pins in port A

Table 17-17. 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 = 00000000h]

GIODOUTA is shown in Table 17-18.

Return to the Summary Table.

GIO data output for pins in port A

Table 17-18. 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 = 00000000h]

GIOSETA is shown in Table 17-19.

Return to the Summary Table.

GIO data set for port A

Table 17-19. 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 = 00000000h]

GIOCLRA is shown in Table 17-20.

Return to the Summary Table.

GIO data clear for port A

Table 17-20. 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 = 00000000h]

GIOPDRA is shown in Table 17-21.

Return to the Summary Table.

GIO open drain for port A

Table 17-21. 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 = 00000000h]

GIOPULDISA is shown in Table 17-22.

Return to the Summary Table.

GIO pul disable for port A

Table 17-22. 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 = 00000000h]

GIOPSLA is shown in Table 17-23.

Return to the Summary Table.

GIO pul select for port A

Table 17-23. 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 = 00000000h]

GIODIRB is shown in Table 17-24.

Return to the Summary Table.

GIO data direction of pins in Port B

Table 17-24. 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 = 00000000h]

GIODINB is shown in Table 17-25.

Return to the Summary Table.

GIO data input for pins in port B

Table 17-25. 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 = 00000000h]

GIODOUTB is shown in Table 17-26.

Return to the Summary Table.

GIO data output for pins in port B

Table 17-26. 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 = 00000000h]

GIOSETB is shown in Table 17-27.

Return to the Summary Table.

GIO data set for port B

Table 17-27. 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 = 00000000h]

GIOCLRB is shown in Table 17-28.

Return to the Summary Table.

GIO data clear for port B

Table 17-28. 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 = 00000000h]

GIOPDRB is shown in Table 17-29.

Return to the Summary Table.

GIO open drain for port B

Table 17-29. 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 = 00000000h]

GIOPULDISB is shown in Table 17-30.

Return to the Summary Table.

GIO pul disable for port B

Table 17-30. 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 = 00000000h]

GIOPSLB is shown in Table 17-31.

Return to the Summary Table.

GIO pul select for port B

Table 17-31. 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 = 00000000h]

GIODIRC is shown in Table 17-32.

Return to the Summary Table.

GIO data direction of pins in Port C

Table 17-32. 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 = 00000000h]

GIODINC is shown in Table 17-33.

Return to the Summary Table.

GIO data input for pins in port C

Table 17-33. 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 = 00000000h]

GIODOUTC is shown in Table 17-34.

Return to the Summary Table.

GIO data output for pins in port C

Table 17-34. 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 = 00000000h]

GIOSETC is shown in Table 17-35.

Return to the Summary Table.

GIO data set for port C

Table 17-35. 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 = 00000000h]

GIOCLRC is shown in Table 17-36.

Return to the Summary Table.

GIO data clear for port C

Table 17-36. 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 = 00000000h]

GIOPDRC is shown in Table 17-37.

Return to the Summary Table.

GIO open drain for port C

Table 17-37. 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 = 00000000h]

GIOPULDISC is shown in Table 17-38.

Return to the Summary Table.

GIO pul disable for port C

Table 17-38. 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 = 00000000h]

GIOPSLC is shown in Table 17-39.

Return to the Summary Table.

GIO pul select for port C

Table 17-39. 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 = 00000000h]

GIODIRD is shown in Table 17-40.

Return to the Summary Table.

GIO data direction of pins in Port D

Table 17-40. 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 = 00000000h]

GIODIND is shown in Table 17-41.

Return to the Summary Table.

GIO data input for pins in port D

Table 17-41. 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 = 00000000h]

GIODOUTD is shown in Table 17-42.

Return to the Summary Table.

GIO data output for pins in port D

Table 17-42. 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 = 00000000h]

GIOSETD is shown in Table 17-43.

Return to the Summary Table.

GIO data set for port D

Table 17-43. 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 = 00000000h]

GIOCLRD is shown in Table 17-44.

Return to the Summary Table.

GIO data clear for port D

Table 17-44. 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 = 00000000h]

GIOPDRD is shown in Table 17-45.

Return to the Summary Table.

GIO open drain for port D

Table 17-45. 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 = 00000000h]

GIOPULDISD is shown in Table 17-46.

Return to the Summary Table.

GIO pul disable for port D

Table 17-46. 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 = 00000000h]

GIOPSLD is shown in Table 17-47.

Return to the Summary Table.

GIO pul select for port D

Table 17-47. 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 = 00000000h]

GIODIRE is shown in Table 17-48.

Return to the Summary Table.

GIO data direction of pins in Port E

Table 17-48. 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 = 00000000h]

GIODINE is shown in Table 17-49.

Return to the Summary Table.

GIO data input for pins in port E

Table 17-49. 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 = 00000000h]

GIODOUTE is shown in Table 17-50.

Return to the Summary Table.

GIO data output for pins in port E

Table 17-50. 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 = 00000000h]

GIOSETE is shown in Table 17-51.

Return to the Summary Table.

GIO data set for port E

Table 17-51. 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 = 00000000h]

GIOCLRE is shown in Table 17-52.

Return to the Summary Table.

GIO data clear for port E

Table 17-52. 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 = 00000000h]

GIOPDRE is shown in Table 17-53.

Return to the Summary Table.

GIO open drain for port E

Table 17-53. 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 = 00000000h]

GIOPULDISE is shown in Table 17-54.

Return to the Summary Table.

GIO pul disable for port E

Table 17-54. 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 = 00000000h]

GIOPSLE is shown in Table 17-55.

Return to the Summary Table.

GIO pul select for port E

Table 17-55. 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 = 00000000h]

GIODIRF is shown in Table 17-56.

Return to the Summary Table.

GIO data direction of pins in Port F

Table 17-56. 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 = 00000000h]

GIODINF is shown in Table 17-57.

Return to the Summary Table.

GIO data input for pins in Port F

Table 17-57. 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 = 00000000h]

GIODOUTF is shown in Table 17-58.

Return to the Summary Table.

GIO data output for pins in Port F

Table 17-58. 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 = 00000000h]

GIOSETF is shown in Table 17-59.

Return to the Summary Table.

GIO data set for Port F

Table 17-59. 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 = 00000000h]

GIOCLRF is shown in Table 17-60.

Return to the Summary Table.

GIO data clear for Port F

Table 17-60. 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 = 00000000h]

GIOPDRF is shown in Table 17-61.

Return to the Summary Table.

GIO open drain for Port F

Table 17-61. 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 = 00000000h]

GIOPULDISF is shown in Table 17-62.

Return to the Summary Table.

GIO pul disable for port F

Table 17-62. 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 = 00000000h]

GIOPSLF is shown in Table 17-63.

Return to the Summary Table.

GIO pul select for port F

Table 17-63. 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 = 00000000h]

GIODIRG is shown in Table 17-64.

Return to the Summary Table.

GIO data direction of pins in Port G

Table 17-64. 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 = 00000000h]

GIODING is shown in Table 17-65.

Return to the Summary Table.

GIO data input for pins in port G

Table 17-65. 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 = 00000000h]

GIODOUTG is shown in Table 17-66.

Return to the Summary Table.

GIO data output for pins in port G

Table 17-66. 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 = 00000000h]

GIOSETG is shown in Table 17-67.

Return to the Summary Table.

GIO data set for port G

Table 17-67. 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 = 00000000h]

GIOCLRG is shown in Table 17-68.

Return to the Summary Table.

GIO data clear for port G

Table 17-68. 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 = 00000000h]

GIOPDRG is shown in Table 17-69.

Return to the Summary Table.

GIO open drain for port G

Table 17-69. 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 = 00000000h]

GIOPULDISG is shown in Table 17-70.

Return to the Summary Table.

GIO pul disable for port G

Table 17-70. 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 = 00000000h]

GIOPSLG is shown in Table 17-71.

Return to the Summary Table.

GIO pul select for port G

Table 17-71. 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 = 00000000h]

GIODIRH is shown in Table 17-72.

Return to the Summary Table.

GIO data direction of pins in Port H

Table 17-72. 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 = 00000000h]

GIODINH is shown in Table 17-73.

Return to the Summary Table.

GIO data input for pins in Port H

Table 17-73. 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 = 00000000h]

GIODOUTH is shown in Table 17-74.

Return to the Summary Table.

GIO data output for pins in Port H

Table 17-74. 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 = 00000000h]

GIOSETH is shown in Table 17-75.

Return to the Summary Table.

GIO data set for Port H

Table 17-75. 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 = 00000000h]

GIOCLRH is shown in Table 17-76.

Return to the Summary Table.

GIO data clear for Port H

Table 17-76. 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 = 00000000h]

GIOPDRH is shown in Table 17-77.

Return to the Summary Table.

GIO open drain for Port H

Table 17-77. 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 = 00000000h]

GIOPULDISH is shown in Table 17-78.

Return to the Summary Table.

GIO pul disable for port H

Table 17-78. 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 = 00000000h]

GIOPSLH is shown in Table 17-79.

Return to the Summary Table.

GIO pul select for port H

Table 17-79. 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 = 00000000h]

GIOSRCA is shown in Table 17-80.

Return to the Summary Table.

GIO slew rate select for port A

Table 17-80. 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 = 00000000h]

GIOSRCB is shown in Table 17-81.

Return to the Summary Table.

GIO slew rate select for port B

Table 17-81. 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 = 00000000h]

GIOSRCC is shown in Table 17-82.

Return to the Summary Table.

GIO slew rate select for port C

Table 17-82. 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 = 00000000h]

GIOSRCD is shown in Table 17-83.

Return to the Summary Table.

GIO slew rate select for port D

Table 17-83. 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 = 00000000h]

GIOSRCE is shown in Table 17-84.

Return to the Summary Table.

GIO slew rate select for port E

Table 17-84. 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 = 00000000h]

GIOSRCF is shown in Table 17-85.

Return to the Summary Table.

GIO slew rate select for port F

Table 17-85. 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 = 00000000h]

GIOSRCG is shown in Table 17-86.

Return to the Summary Table.

GIO slew rate select for port G

Table 17-86. 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 = 00000000h]

GIOSRCH is shown in Table 17-87.

Return to the Summary Table.

GIO slew rate select for port H

Table 17-87. 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 17-88.

Table 17-88. Output Buffer and Pull Control Behavior for GIO Pins

Module under Reset?	Pin Direction (GIODIR)(1) (2)	Open Drain Enable (GIOPDR)(1)	Pull Disable (GIOPULDIS)(1) (3)	Pull Select (GIOPSL)(1) (4)	Pull Control	Output Buffer(5)
Yes	X	X	X	X	Enabled	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) GIOPULDIS = 0 for enabling pull control; = 1 for disabling pull control
(4) GIOPSL= 0 for pull-down functionality; = 1 for pull-up functionality
(5) If open drain is enabled, output buffer will be disabled if a high level (1) is being output.

18 Enhanced Pulse Width Modulator (ePWM) Module

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.

18.1 Introduction

An effective PWM peripheral must be able to generate complex pulse width waveforms with minimal CPU overhead or intervention. It needs to be highly programmable and very flexible while being easy to understand and use. The ePWM unit described here addresses these requirements by allocating all needed timing and control resources on a per PWM channel basis. Cross coupling or sharing of resources has been avoided; instead, 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.

In this document 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 EPWM4A and EPWM4B belong to ePWM4.

18.1.1 Submodule Overview

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 Figure 18-1. 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. Additionally, this synchronization scheme can be extended to the capture peripheral modules (eCAP). Modules can also operate stand-alone.

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.

Each ePWM module is connected to the input/output signals shown in Figure 18-1. The signals are described in detail in subsequent sections.

Each ePWM module consists of eight submodules and is connected within a system via the signals shown in Figure 18-2.

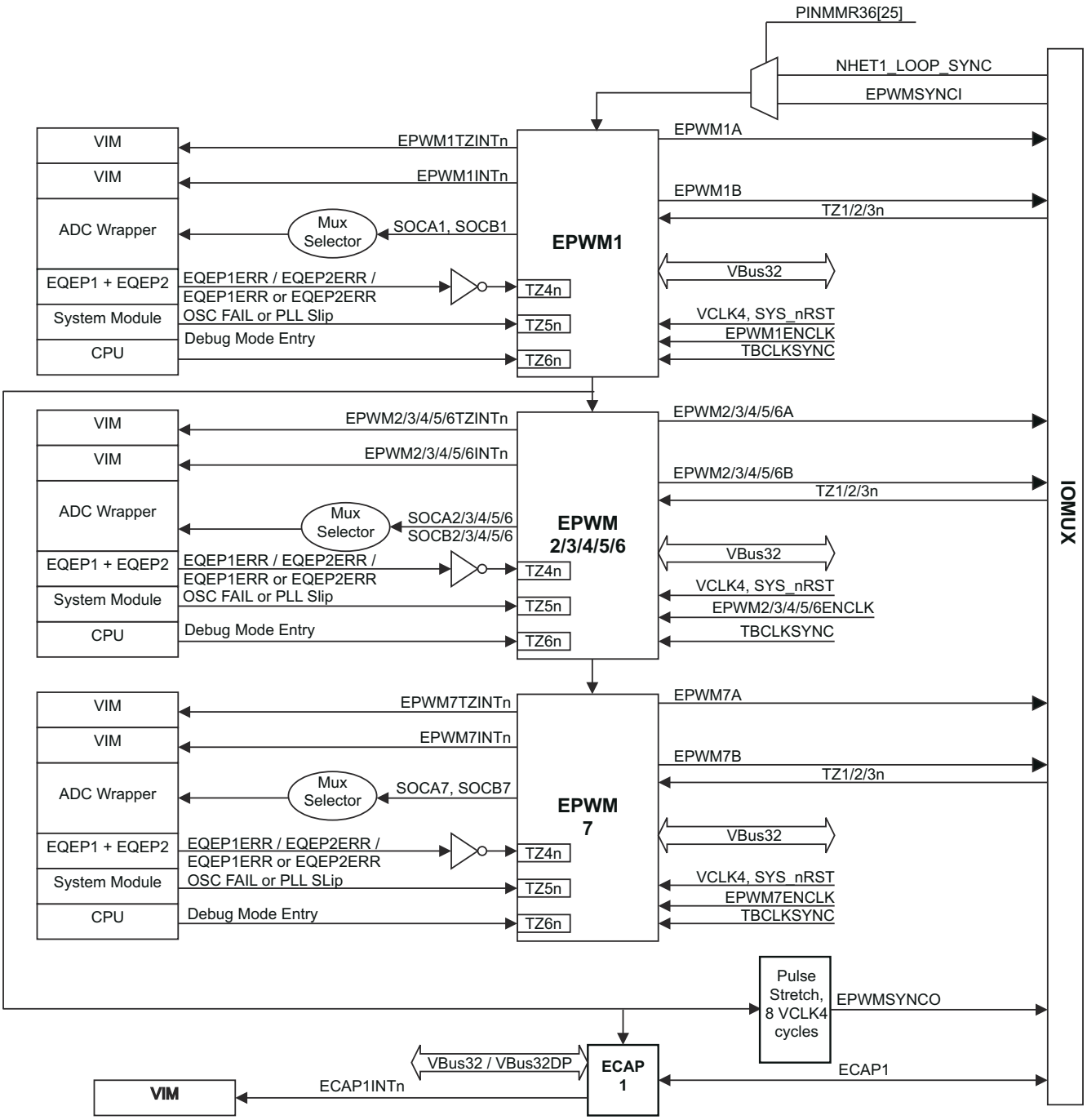


Figure 18-1. Multiple ePWM Modules

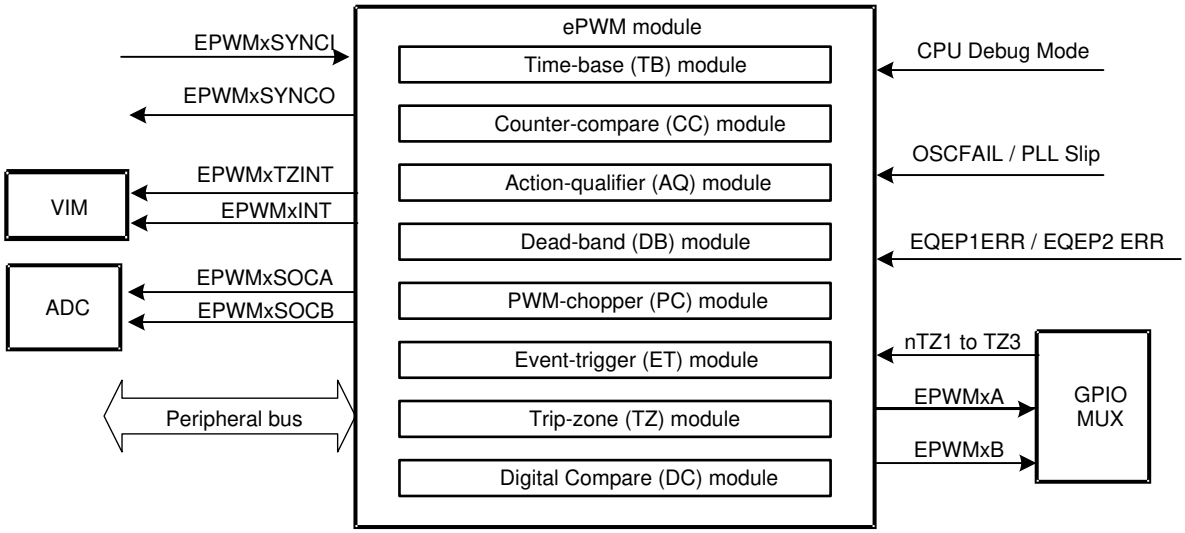


Figure 18-2. Submodules and Signal Connections for an ePWM Module

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 of the device's technical reference manual.
- **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 VCLK4, or double-synchronized and filtered through a 6-VCLK4-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.

ADVANCE INFORMATION

18.1.2 Register Mapping

The complete ePWM module control and status register set is grouped by submodule as shown in Table 18-1. Each register set is duplicated for each instance of the ePWM module. The start address for each ePWM register file instance on a device is specified in the specific part's datasheet.

Table 18-1. ePWM Module Control and Status Register Set Grouped by Submodule

Name	Address Offset[1]	Size (×16)	Shadow	Privileged Mode Write Only?	Description
Time-Base Submodule Registers					
TBCTL	0x0002	1	No	No	Time-Base Control Register
TBSTS	0x0000	1	No	No	Time-Base Status Register
Reserved	0x0006	1	–	–	Reserved
TBPHS	0x0004	1	No	No	Time-Base Phase Register
TBCTR	0x000A	1	No	No	Time-Base Counter Register
TBPRD	0x0008	1	Yes	No	Time-Base Period Register
Reserved	0x000E	1	–	–	Reserved
Counter-Compare Submodule Registers					
CMPCTL	0x000C	1	No	No	Counter-Compare Control Register
Reserved	0x0012	1	–	–	Reserved
CMPA	0x0010	1	Yes	No	Counter-Compare A Register
CMPB	0x0016	1	Yes	No	Counter-Compare B Register
Action-Qualifier Submodule Registers					
AQCTLA	0x0014	1	No	No	Action-Qualifier Control Register for Output A (EPWMxA)
AQCTLB	0x001A	1	No	No	Action-Qualifier Control Register for Output B (EPWMxB)
AQSFRC	0x0018	1	No	No	Action-Qualifier Software Force Register
AQCSFRC	0x001E	1	Yes	No	Action-Qualifier Continuous S/W Force Register Set
Dead-Band Generator Submodule Registers					
DBCTL	0x001C	1	No	No	Dead-Band Generator Control Register
DBRED	0x0022	1	No	No	Dead-Band Generator Rising Edge Delay Count Register
DBFED	0x0020	1	No	No	Dead-Band Generator Falling Edge Delay Count Register
Trip-Zone Submodule Registers					
TZSEL	0x0026	1	No	Yes	Trip-Zone Select Register
TZDCSEL	0x0024	1	No	Yes	Trip Zone Digital Compare Select Register
TZCTL	0x002A	1	No	Yes	Trip-Zone Control Register
TZEINT	0x0028	1	No	Yes	Trip-Zone Enable Interrupt Register
TZFLG	0x002E	1	No	No	Trip-Zone Flag Register
TZCLR	0x002C	1	No	Yes	Trip-Zone Clear Register
TZFRC	0x0032	1	No	Yes	Trip-Zone Force Register
Event-Trigger Submodule Registers					
ETSEL	0x0030	1	No	No	Event-Trigger Selection Register
ETPS	0x0036	1	No	No	Event-Trigger Pre-Scale Register
ETFLG	0x0034	1	No	No	Event-Trigger Flag Register
ETCLR	0x003A	1	No	No	Event-Trigger Clear Register
ETFRC	0x0038	1	No	No	Event-Trigger Force Register
PWM-Chopper Submodule Registers					
PCCTL	0x003E	1	No	No	PWM-Chopper Control Register

ADVANCE INFORMATION

Table 18-1. ePWM Module Control and Status Register Set Grouped by Submodule (continued)

Name	Address Offset[1]	Size (×16)	Shadow	Privileged Mode Write Only?	Description
Digital Compare Event Registers					
DCTRIPSEL	0x0062	1	No	Yes	Digital Compare Trip Select Register
DCACTL	0x0060	1	No	Yes	Digital Compare A Control Register
DCBCTL	0x0066	1	No	Yes	Digital Compare B Control Register
DCFCTL	0x0064	1	No	Yes	Digital Compare Filter Control Register
DCCAPCTL	0x006A	1	No	Yes	Digital Compare Capture Control Register
DCFOFFSET	0x0068	1	Writes	No	Digital Compare Filter Offset Register
DCFOFFSETCNT	0x006E	1	No	No	Digital Compare Filter Offset Counter Register
DCFWINDOW	0x006C	1	No	No	Digital Compare Filter Window Register
DCFWINDOWCNT	0x0072	1	No	No	Digital Compare Filter Window Counter Register
DCCAP	0x0070	1	Yes	No	Digital Compare Counter Capture Register

(1) Locations not shown are reserved.

18.2 ePWM Submodules

Eight submodules are included in every ePWM peripheral. Each of these submodules performs specific tasks that can be configured by software.

18.2.1 Overview

Table 18-2 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 18.2.3 for relevant details.

Table 18-2. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time-base (TB)	• Scale the time-base clock (TBCLK) relative to the system clock (VCLK4). • 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

Table 18-2. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
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
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 the remainder of this document 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.

18.2.2 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 18-3 illustrates the time-base module's place within the ePWM.

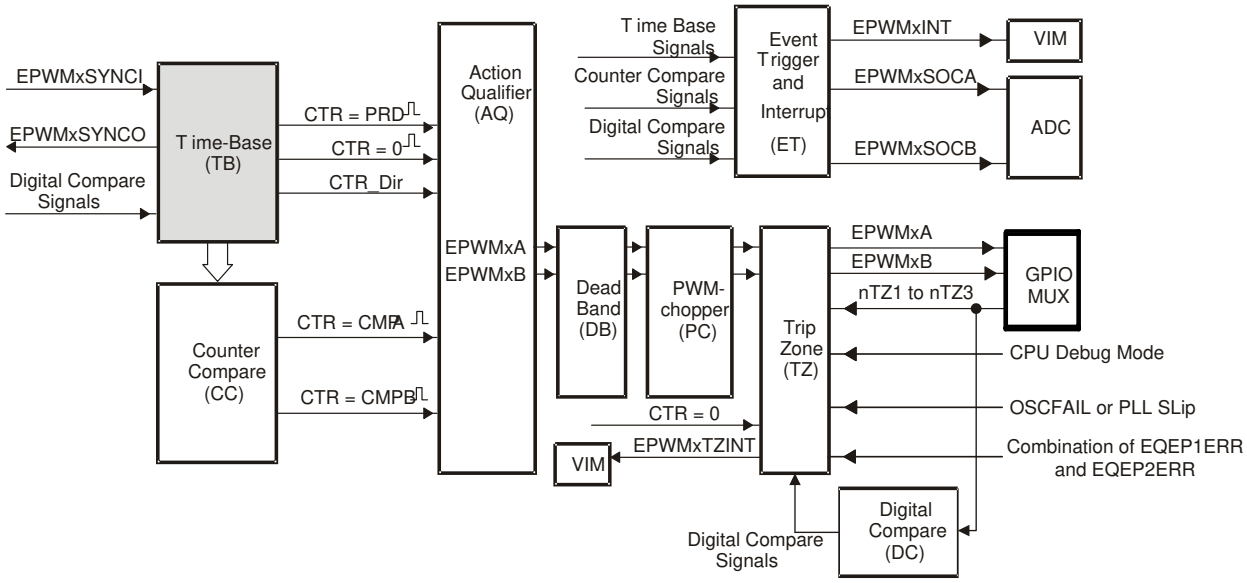


Figure 18-3. Time-Base Submodule Block Diagram

18.2.2.1 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 (VCLK4). This allows the time-base counter to increment/decrement at a slower rate.

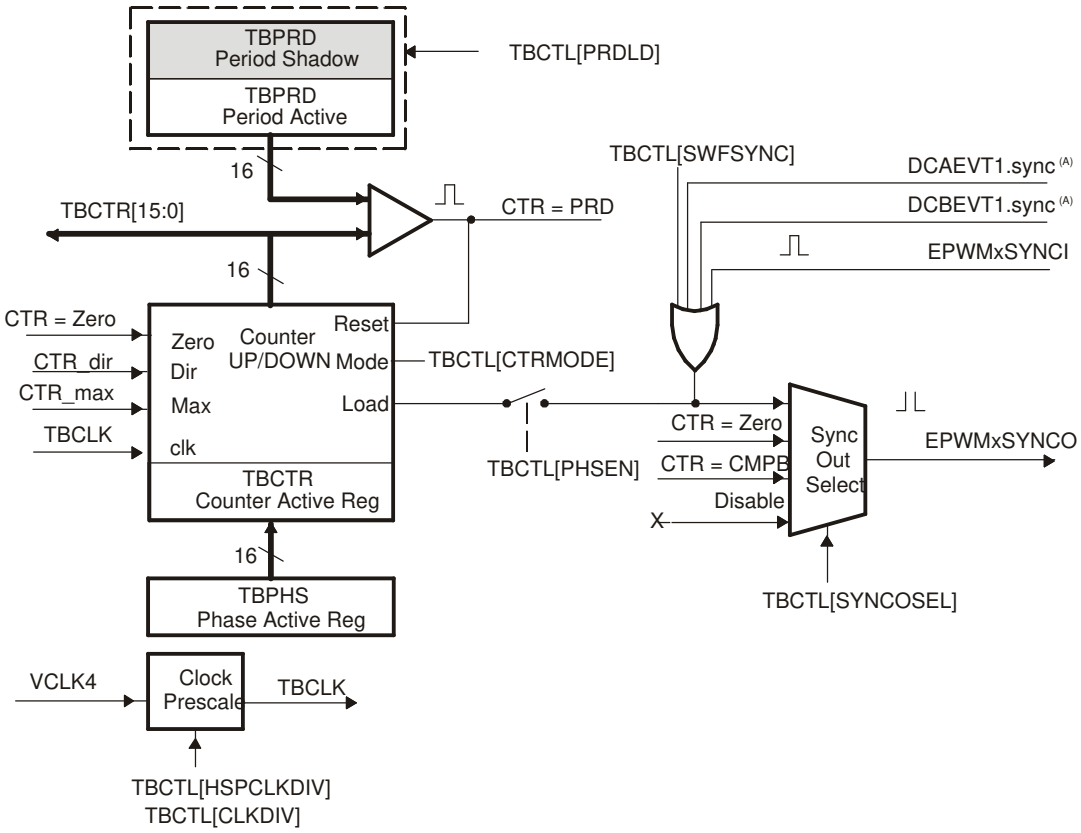
18.2.2.2 Controlling and Monitoring the Time-base Submodule

Table 18-3 shows the registers used to control and monitor the time-base submodule.

Table 18-3. Time-Base Submodule Registers

Register Name	Address Offset	Shadowed	Description
TBCTL	0x0002	No	Time-Base Control Register
TBSTS	0x0000	No	Time-Base Status Register
TBPHS	0x0004	No	Time-Base Phase Register
TBCTR	0x000A	No	Time-Base Counter Register
TBPRD	0x0008	Yes	Time-Base Period Register

The block diagram in Figure 18-4 shows the critical signals and registers of the time-base submodule. Table 18-4 provides descriptions of the key signals associated with the time-base submodule.



A. These signals are generated by the digital compare (DC) submodule.

Figure 18-4. Time-Base Submodule Signals and Registers

Table 18-4. 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 18.2.2.3.3 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.
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 (VCLK4) and is used by all submodules within the ePWM. This clock determines the rate at which time-base counter increments or decrements.

18.2.2.3 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 18-5 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 (VCLK4).

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.

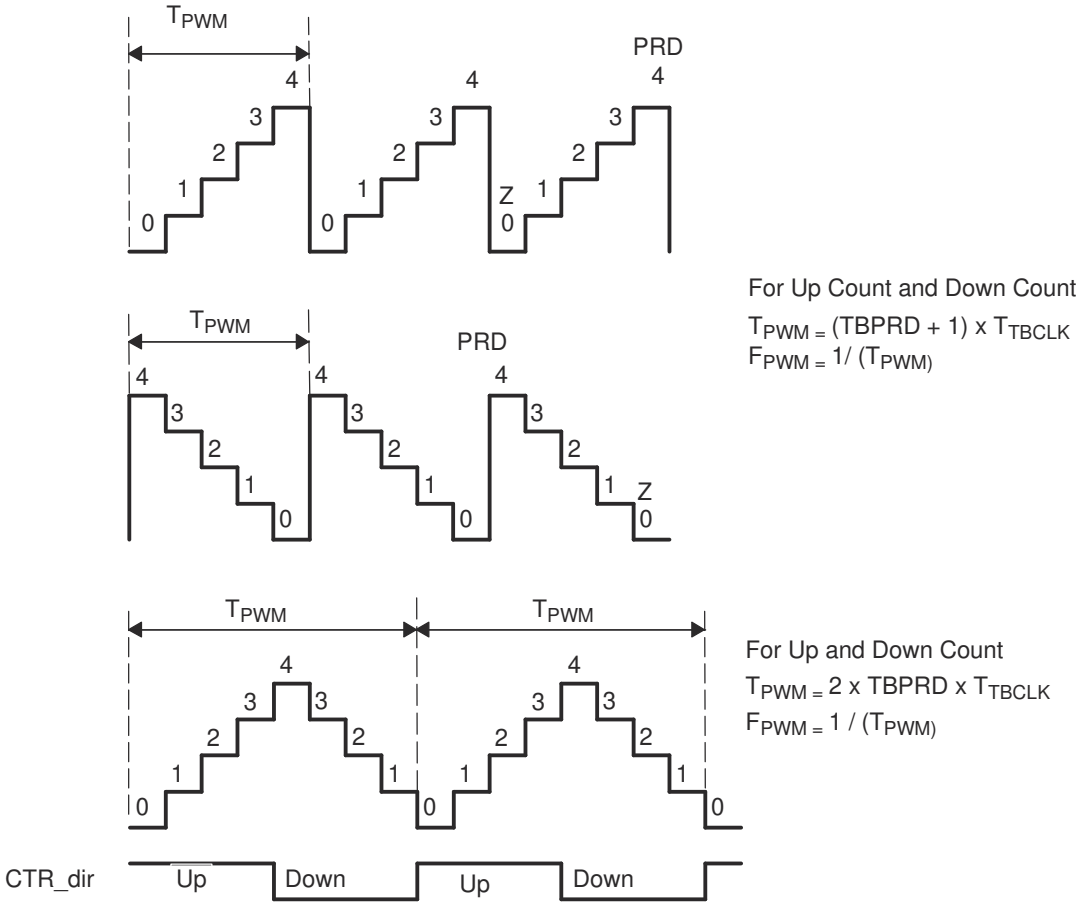


Figure 18-5. Time-Base Frequency and Period

18.2.2.3.1 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.

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.

18.2.2.3.2 Time-Base Clock Synchronization

Bit 1 of the device-level multiplexing control module (IOMM) register PINMMR37 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.

18.2.2.3.3 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 18-6.

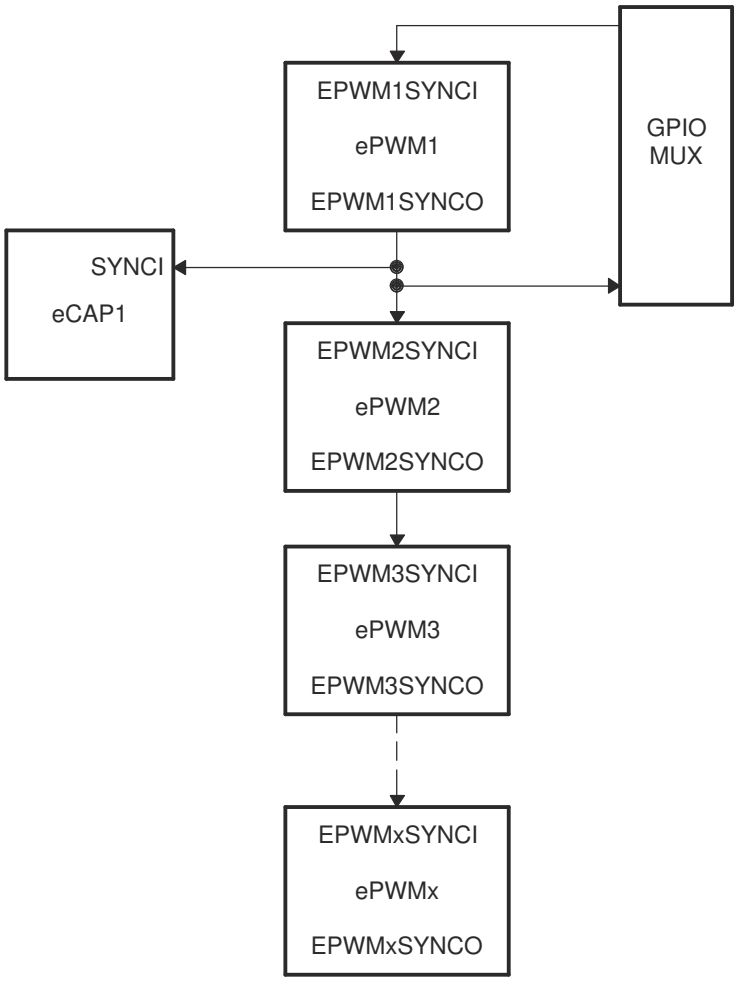


Figure 18-6. Time-Base Counter Synchronization Scheme

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 controller module to target modules is given by:
  - if (TBCLK = VCLK4): 2 × VCLK4
  - if (TBCLK != VCLK4): 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 18-7 through Figure 18-10 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 controller time-base (for example, ePWM1) and downstream modules (ePWM2 - ePWMx) may elect to run in synchronization with the controller.

18.2.2.4 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.

18.2.2.5 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.

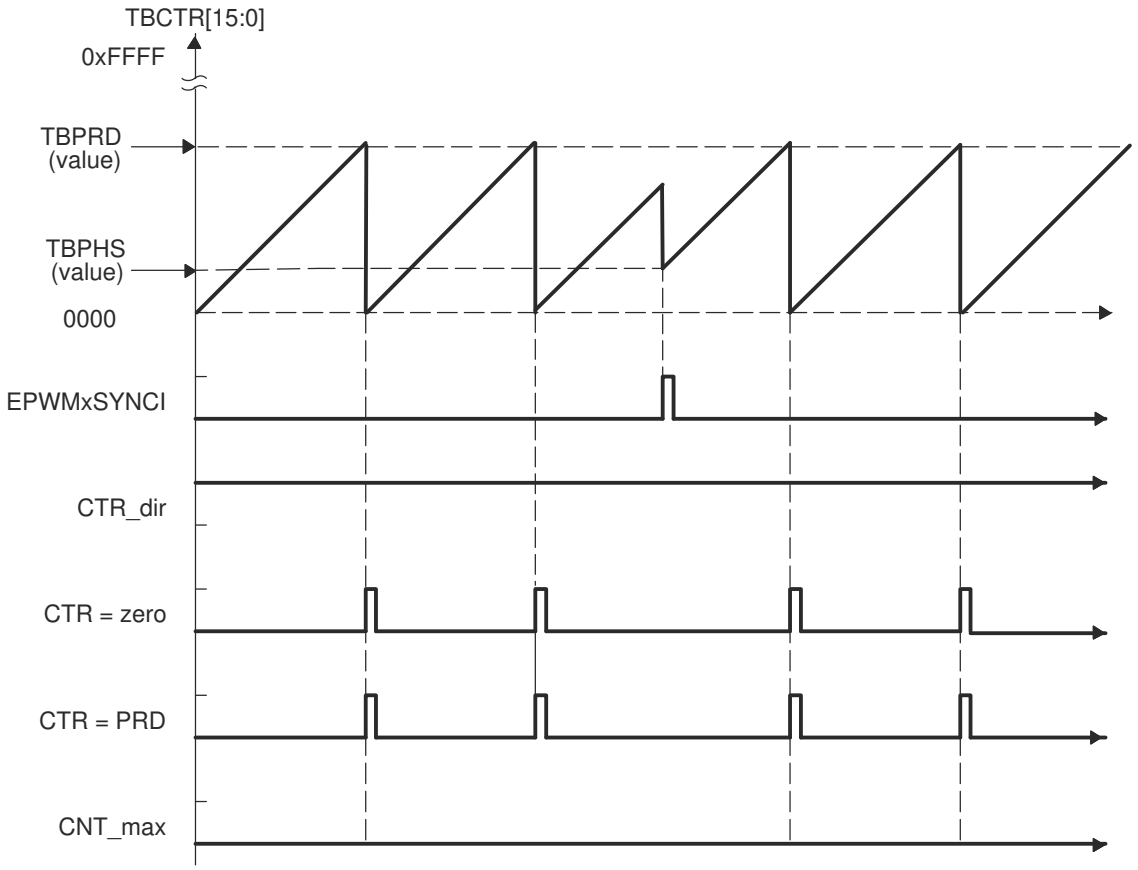


Figure 18-7. Time-Base Up-Count Mode Waveforms

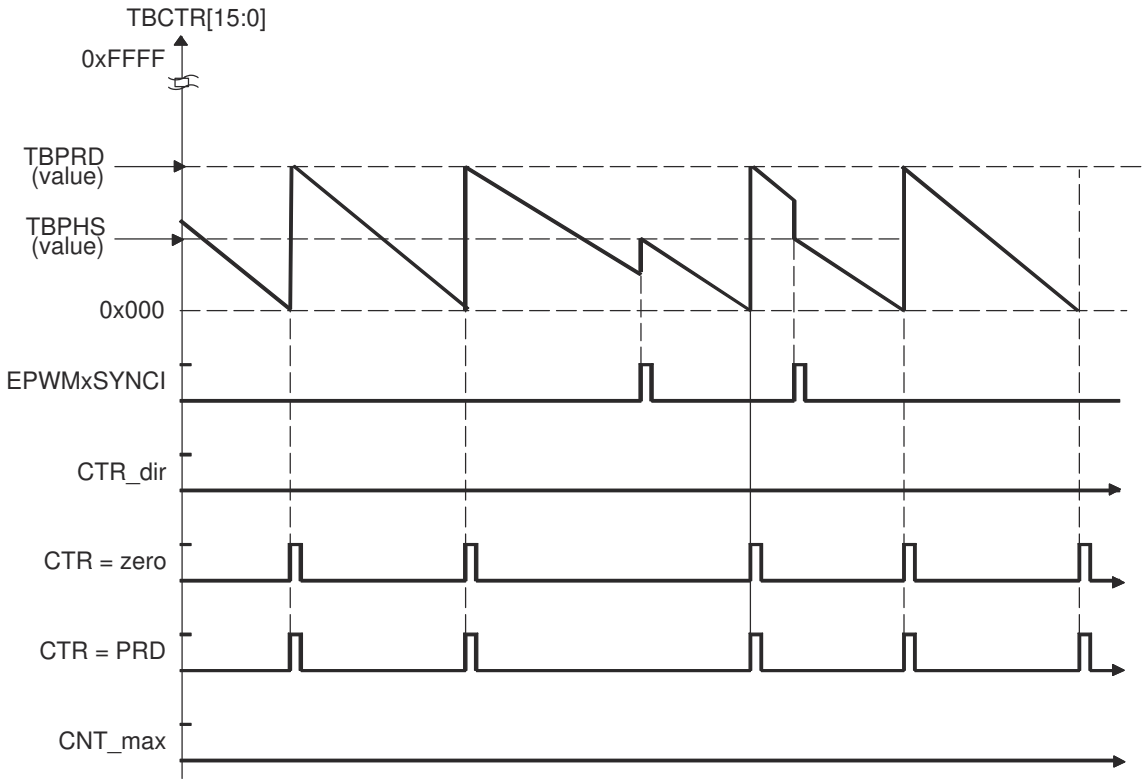


Figure 18-8. Time-Base Down-Count Mode Waveforms

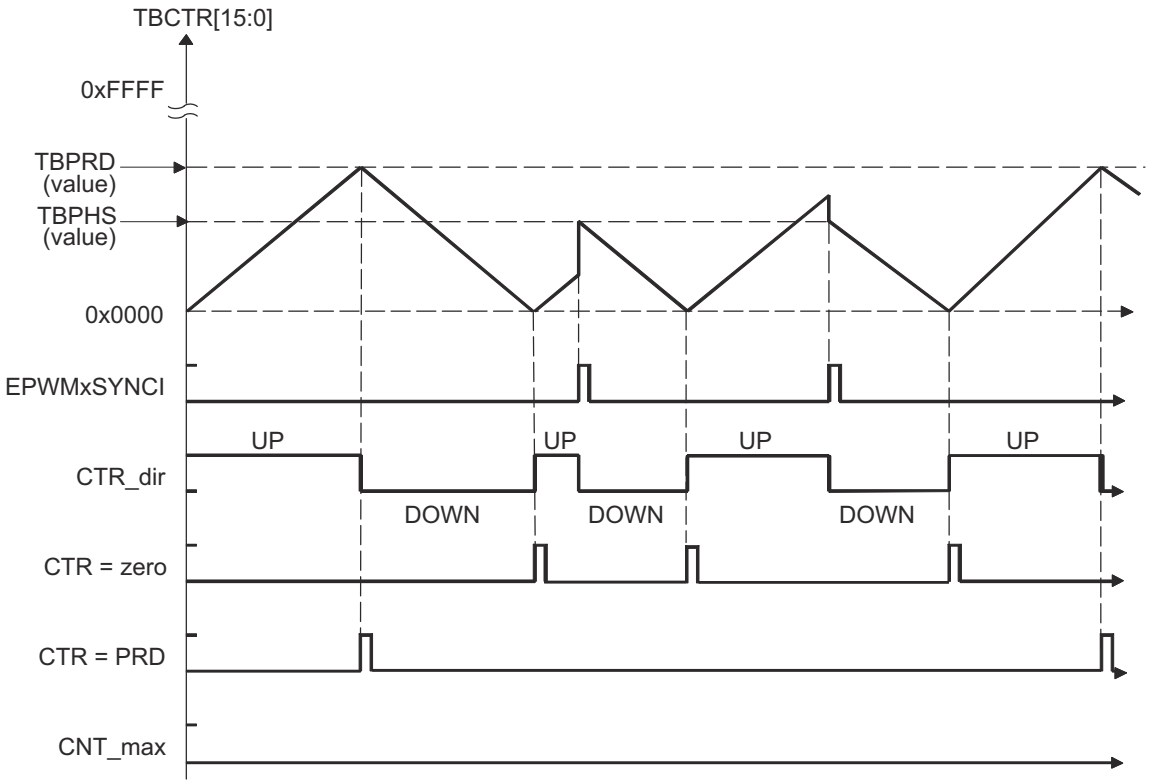


Figure 18-9. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

ADVANCE INFORMATION

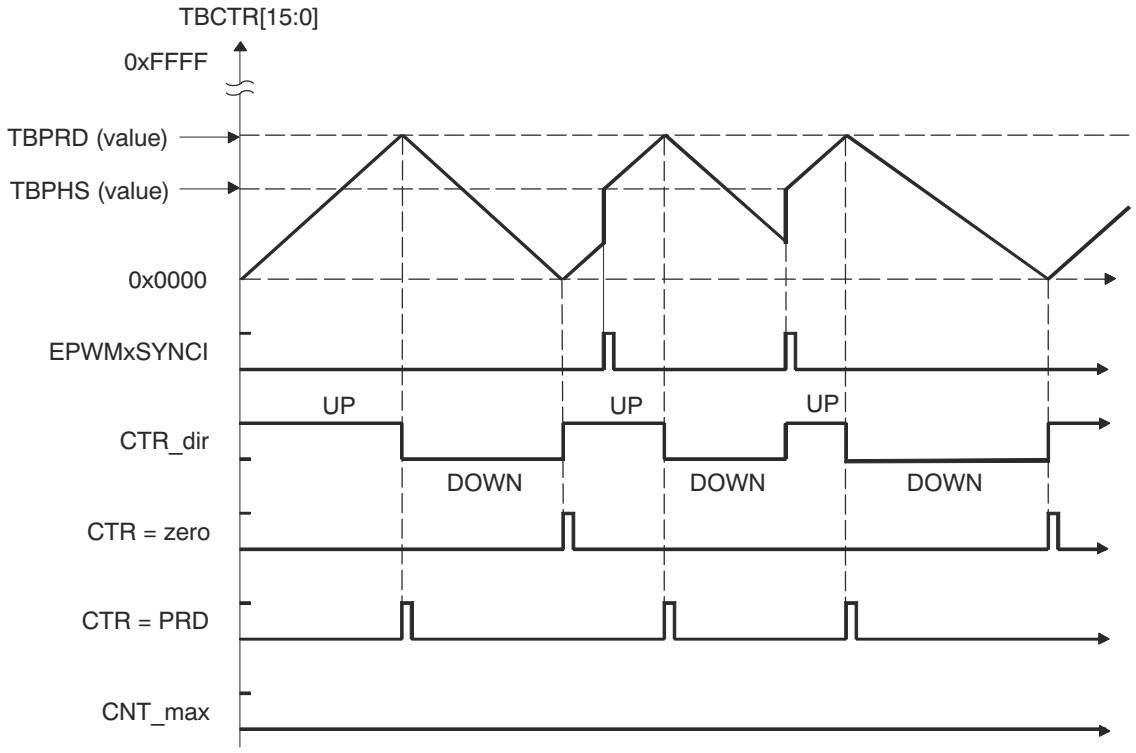


Figure 18-10. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

18.2.3 Counter-Compare (CC) Submodule

Figure 18-11 illustrates the counter-compare submodule within the ePWM.

Figure 18-12 shows the basic structure of the counter-compare submodule.

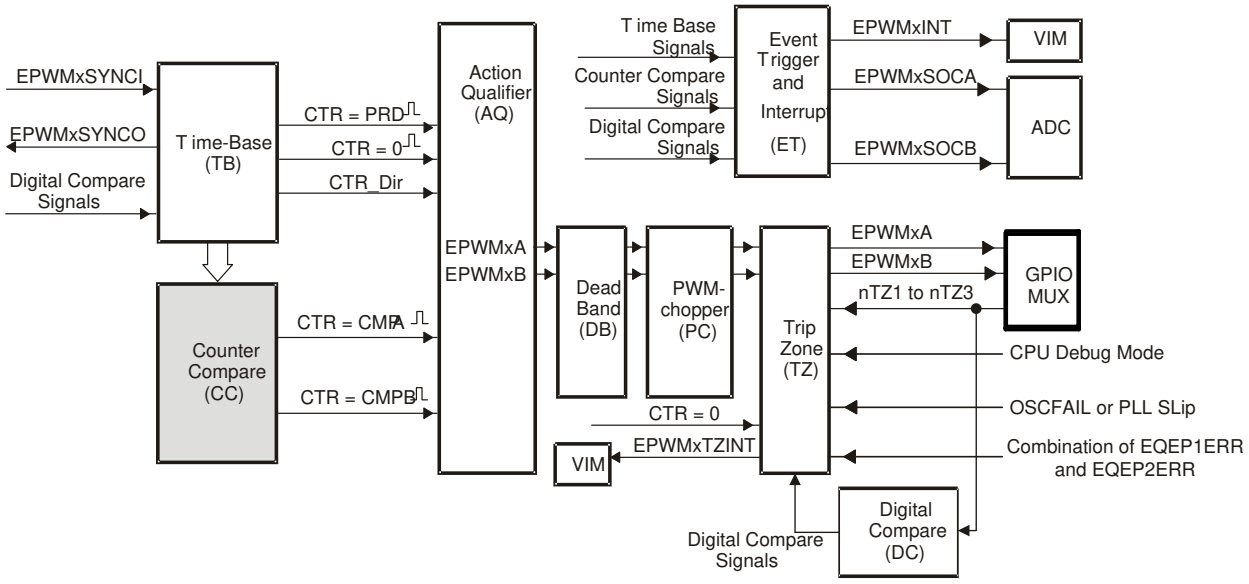


Figure 18-11. Counter-Compare Submodule

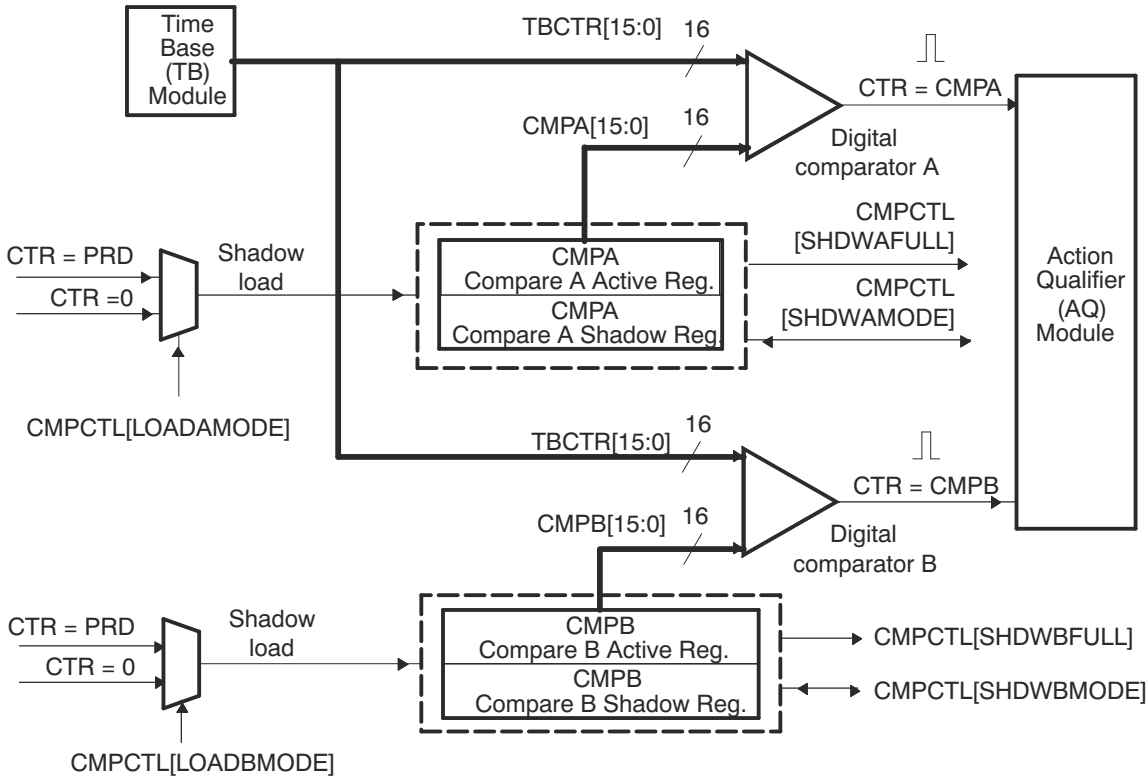


Figure 18-12. Detailed View of the Counter-Compare Submodule

18.2.3.1 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

18.2.3.2 Controlling and Monitoring the Counter-Compare Submodule

The counter-compare submodule operation is controlled and monitored by the registers shown in Table 18-5:

Table 18-5. Counter-Compare Submodule Registers

Register Name	Address Offset	Shadowed	Description
CMPCTL	0x000C	No	Counter-Compare Control Register.
CMPA	0x0010	Yes	Counter-Compare A Register
CMPB	0x0016	Yes	Counter-Compare B Register

The key signals associated with the counter-compare submodule are described in Table 18-6.

Table 18-6. 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

18.2.3.3 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 18.2.4.1 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 as described:

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:

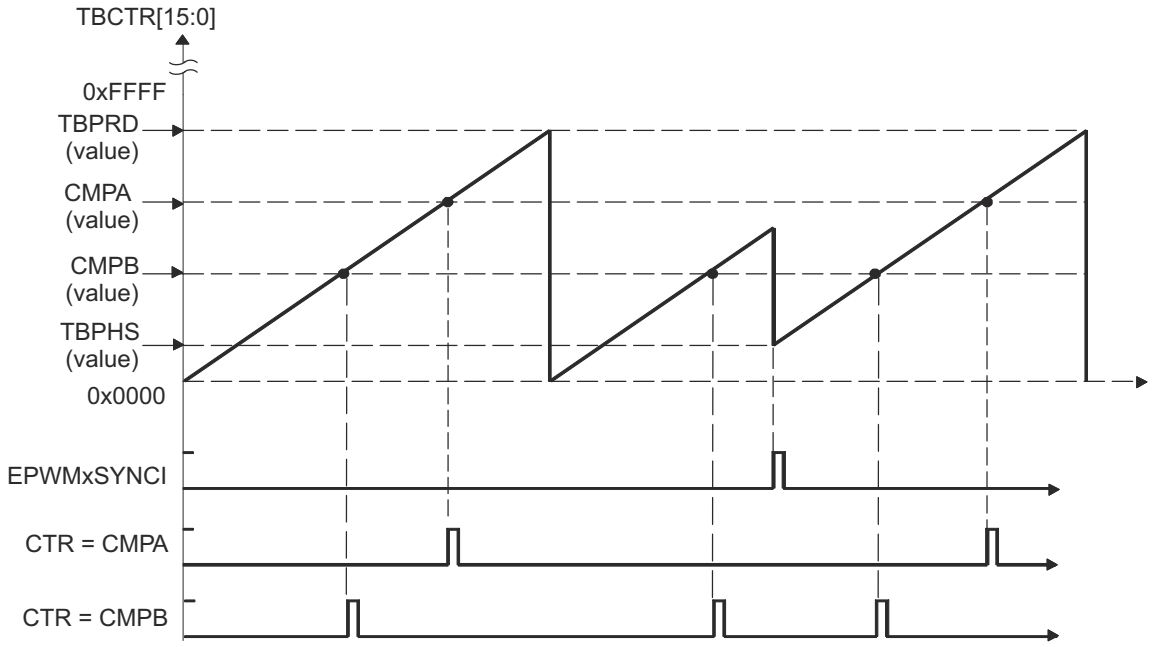
If immediate load mode is selected (that is, TBCTL[SHADWAMODE] = 1 or TBCTL[SHADWBMODE] = 1), then a read from or a write to the register will go directly to the active register.

18.2.3.4 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 best illustrate the operation of the first three modes, the timing diagrams in Figure 18-13 through Figure 18-16 show when events are generated and how the EPWMxSYNCI signal interacts.



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 18-13. Counter-Compare Event Waveforms in Up-Count Mode

ADVANCE INFORMATION

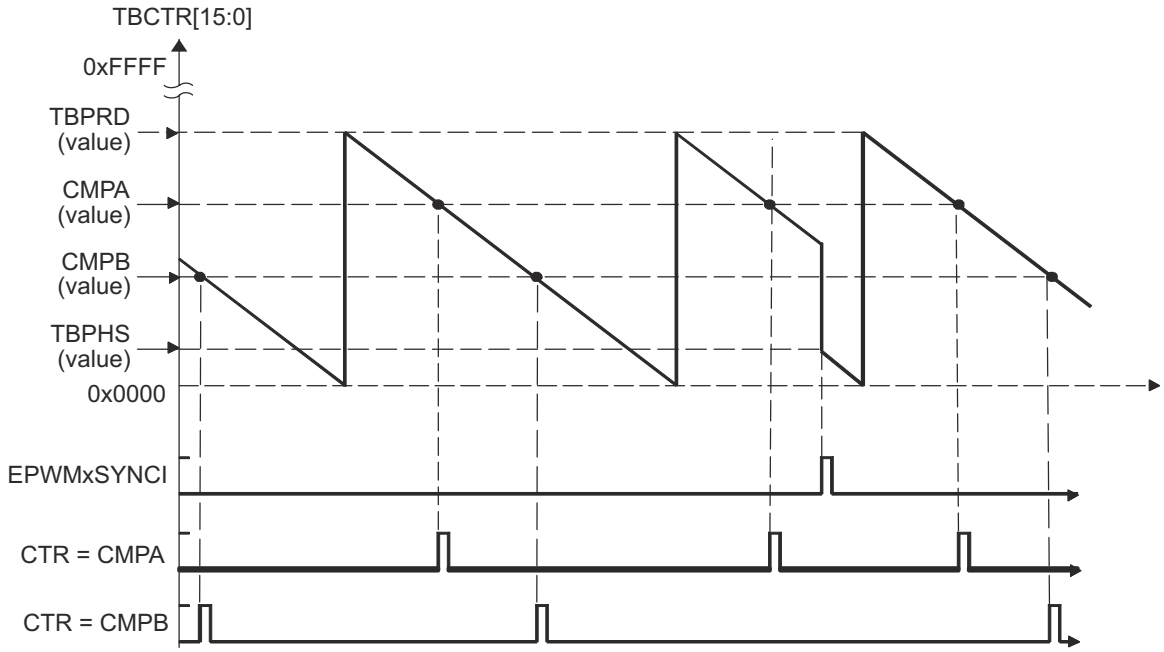


Figure 18-14. Counter-Compare Events in Down-Count Mode

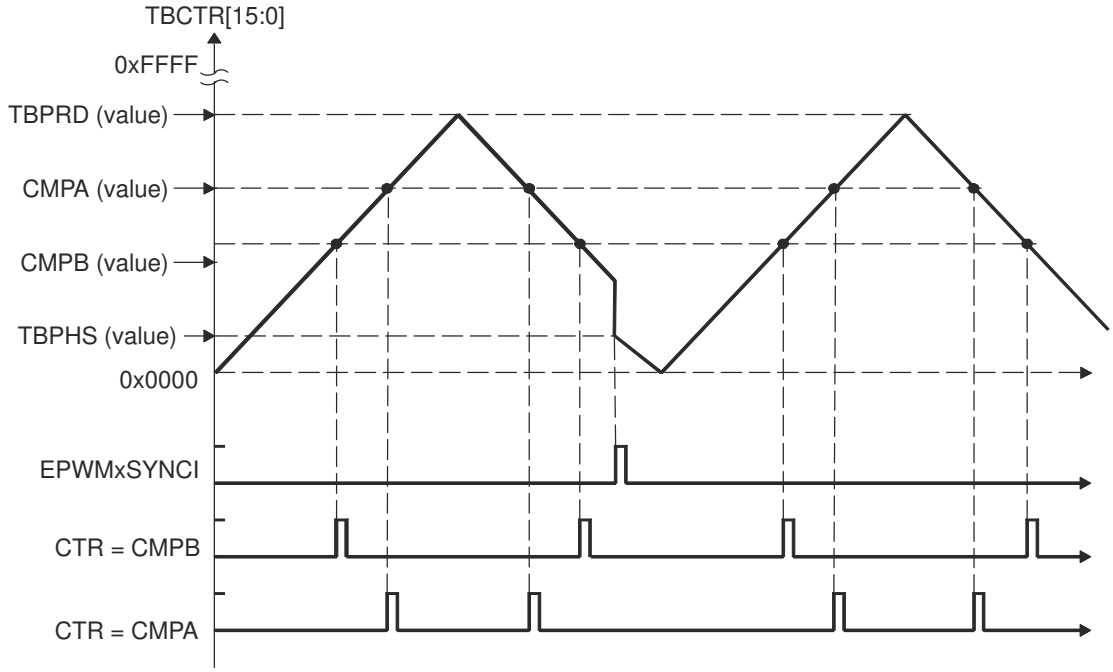


Figure 18-15. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

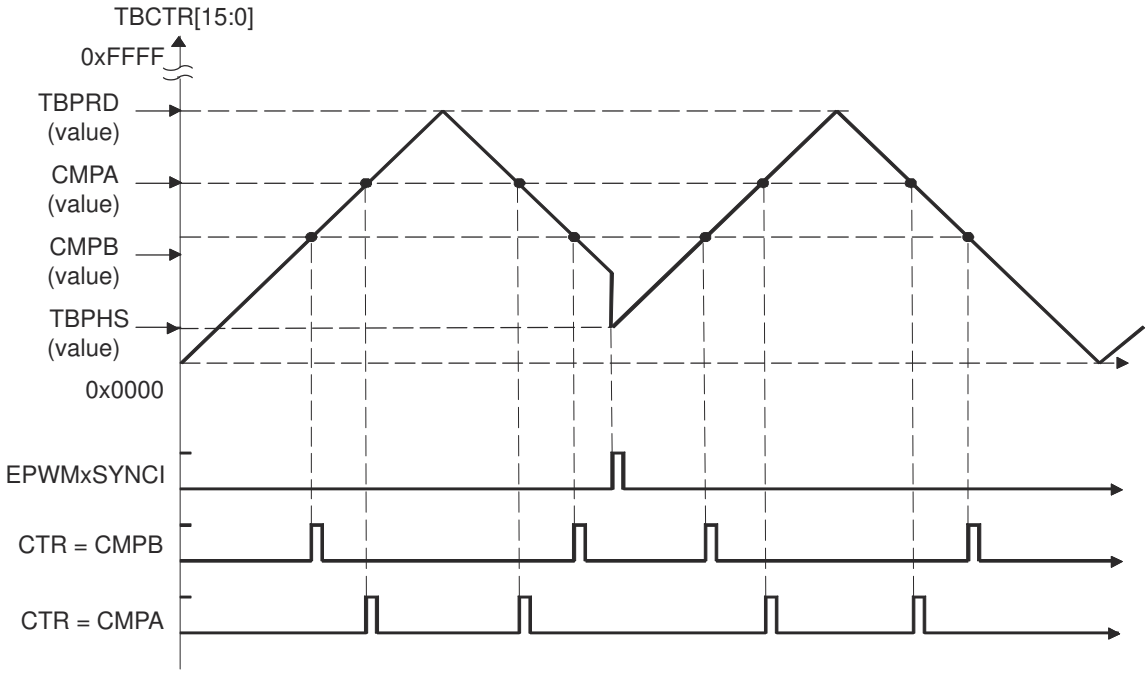


Figure 18-16. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

ADVANCE INFORMATION

18.2.4 Action-Qualifier (AQ) Submodule

Figure 18-17 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.

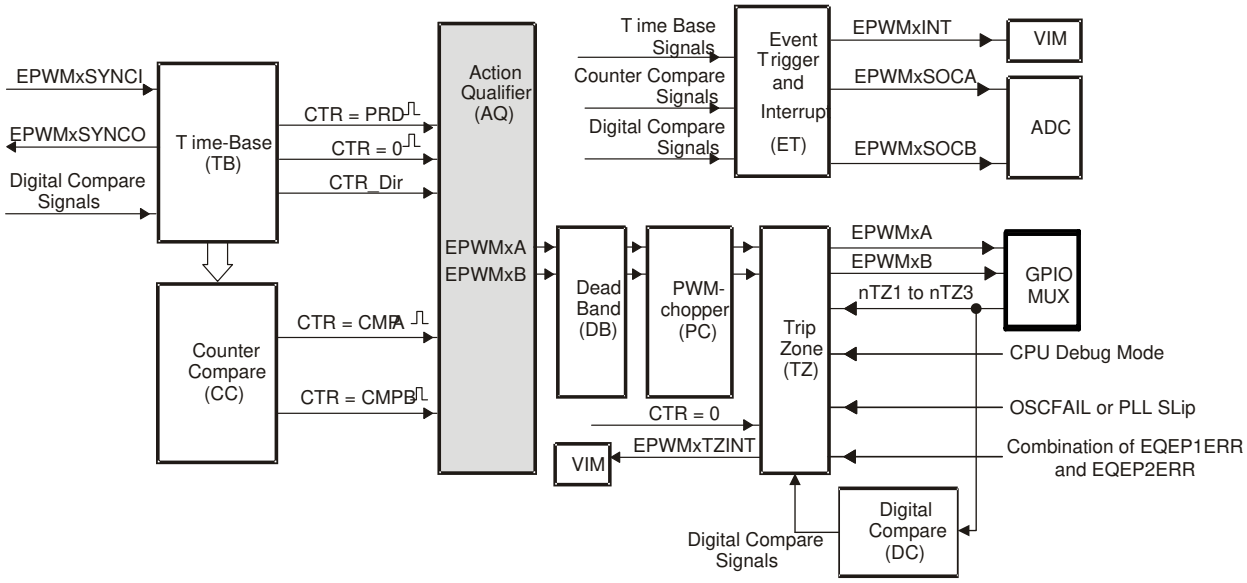


Figure 18-17. Action-Qualifier Submodule

18.2.4.1 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

18.2.4.2 Action-Qualifier Submodule Control and Status Register Definitions

The action-qualifier submodule operation is controlled and monitored via the registers in Table 18-7.

Table 18-7. Action-Qualifier Submodule Registers

Register Name	Address Offset	Shadowed	Description
AQCTLA	0x0014	No	Action-Qualifier Control Register For Output A (EPWMxA)
AQCTLB	0x001A	No	Action-Qualifier Control Register For Output B (EPWMxB)
AQSFRC	0x0018	No	Action-Qualifier Software Force Register
AQCSFRC	0x001E	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 18-7.

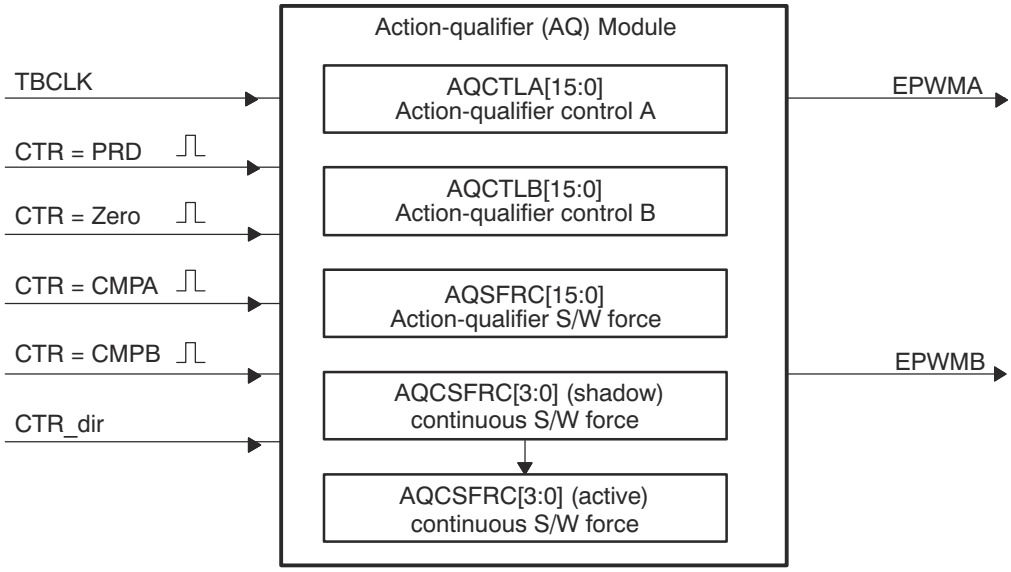


Figure 18-18. Action-Qualifier Submodule Inputs and Outputs

For convenience, the possible input events are summarized again in Table 18-8.

Table 18-8. 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 18.2.8 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 18-19. 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.

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

Figure 18-19. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs

18.2.4.3 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 18-9. 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 18-9. 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 18-10 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 18-10. 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 18-11 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 18-11. 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 18-12.

Table 18-12. 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.
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).

18.2.4.4 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 (SPRAAI1)

Figure 18-20 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.

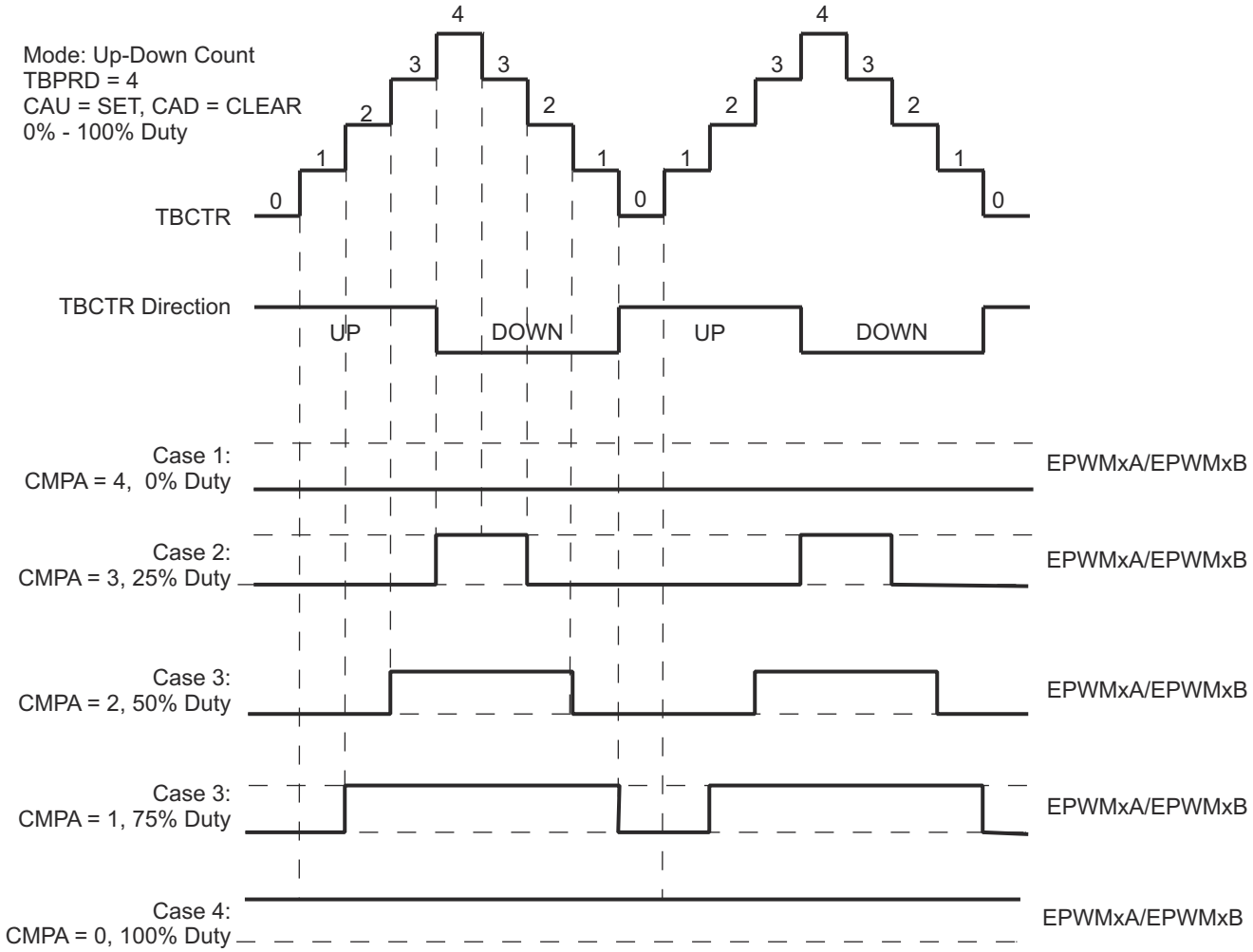
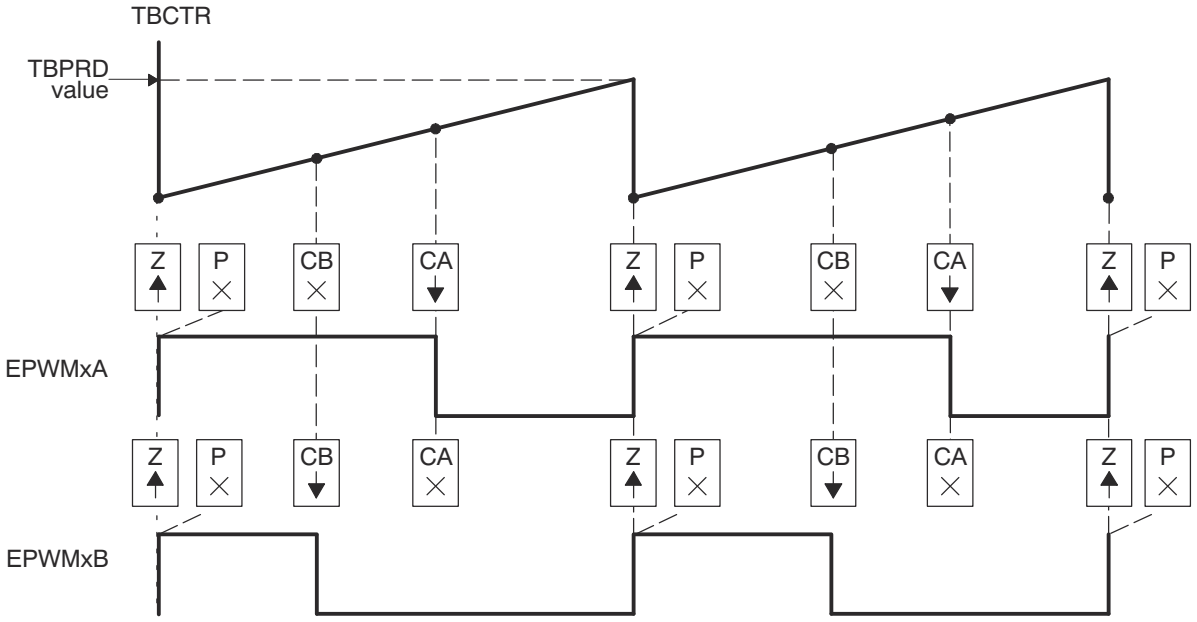


Figure 18-20. Up-Down-Count Mode Symmetrical Waveform

ADVANCE INFORMATION

The PWM waveforms in Figure 18-21 through Figure 18-26 show some common action-qualifier configurations. The C-code samples in Example 18-1 through Example 18-6 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



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.

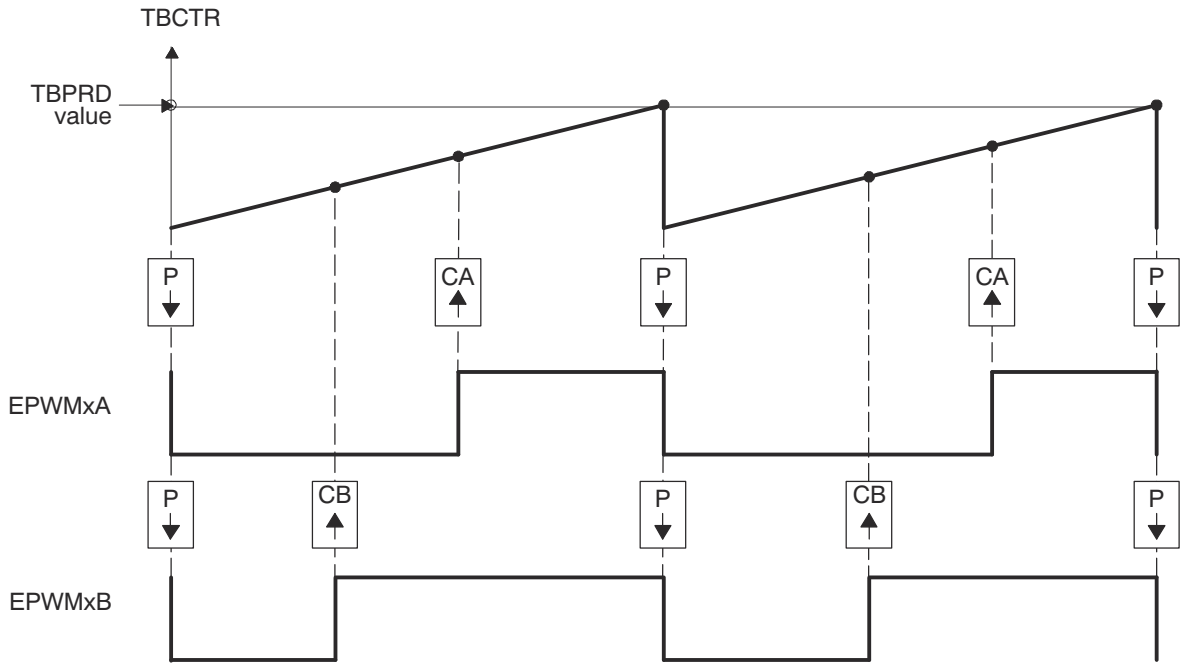
Figure 18-21. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High

Example 18-1 contains a code sample showing initialization and run time for the waveforms in Figure 18-21.

Example 18-1. Code Sample for Figure 18-21

```
// 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 = 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
```

18.2.4.5



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.

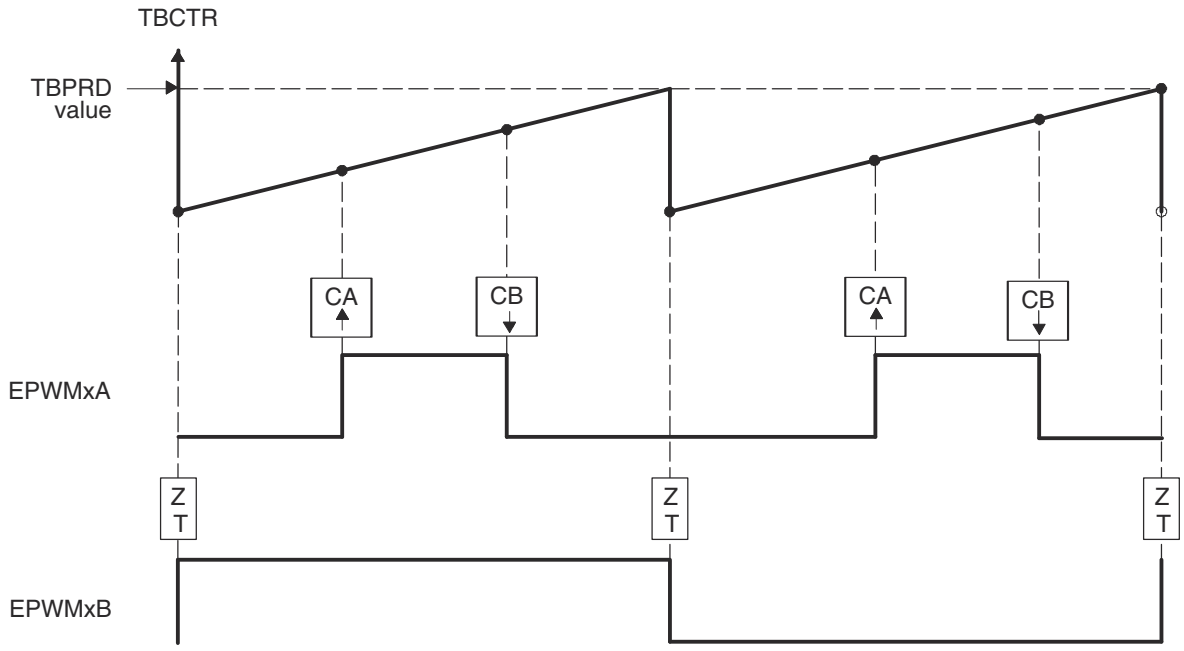
Figure 18-22. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low

Example 18-2 contains a code sample showing initialization and run time for the waveforms in Figure 18-22.

Example 18-2. Code Sample for Figure 18-22

```
// 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 = VCLK4
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
```

18.2.4.6



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)

Figure 18-23. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA

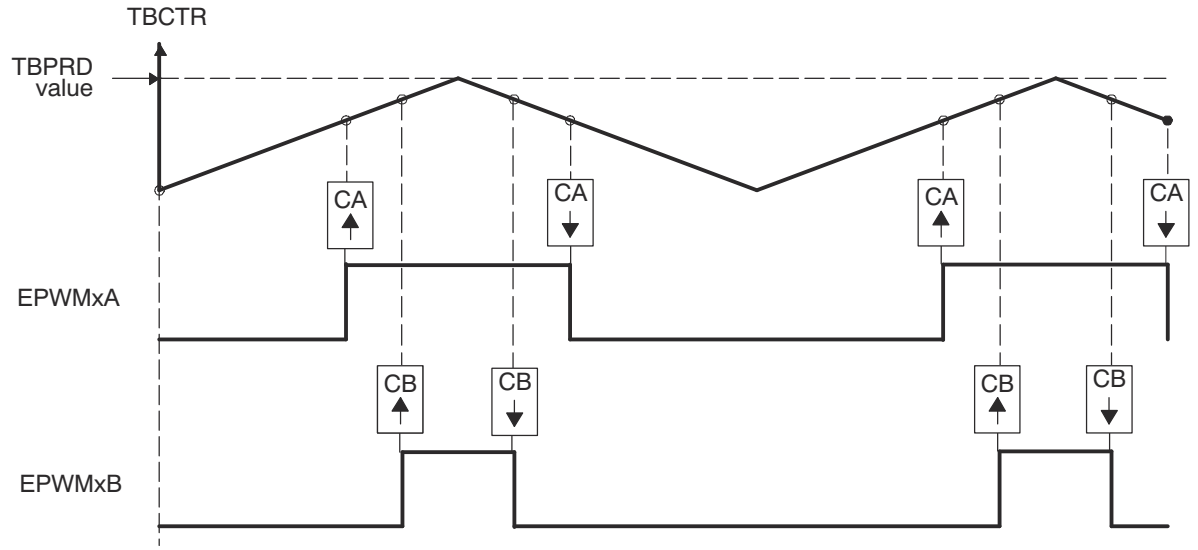
ADVANCE INFORMATION

Example 18-3 contains a code sample showing initialization and run time for the waveforms in Figure 18-23.

Example 18-3. Code Sample for Figure 18-23

```
// 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 = VCLK4
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;
```

18.2.4.7



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

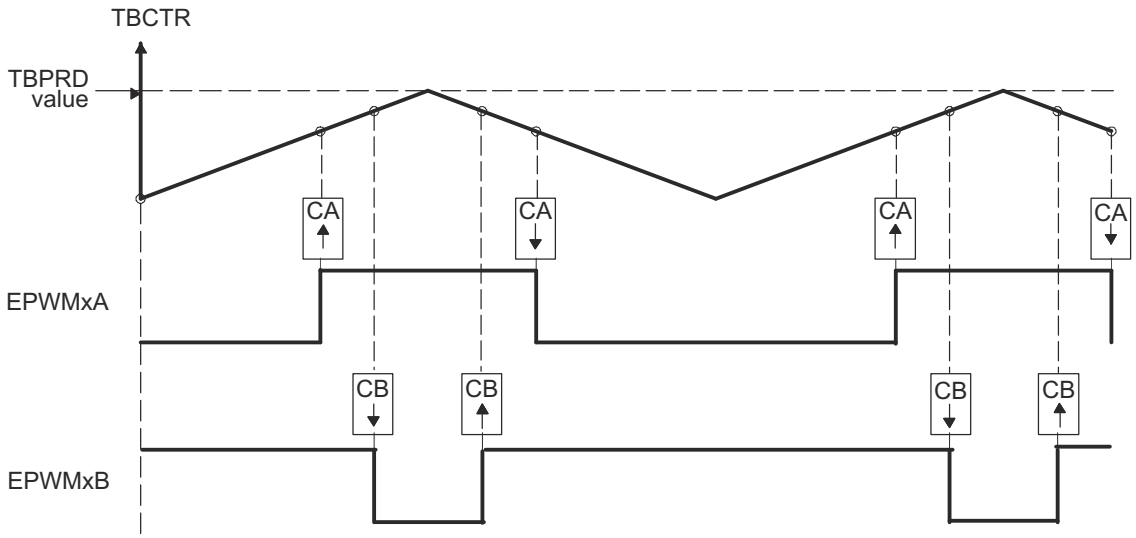
Figure 18-24. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low

Example 18-4 contains a code sample showing initialization and run time for the waveforms in Figure 18-24.

Example 18-4. Code Sample for Figure 18-24

```
// 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 = VCLK4
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
```

18.2.4.8



A. PWM period = 2 × TBPRD × T_{TBCLK}
B. Duty modulation for EPWMxA is set by CMPA, and is active low, that is, low 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. 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.

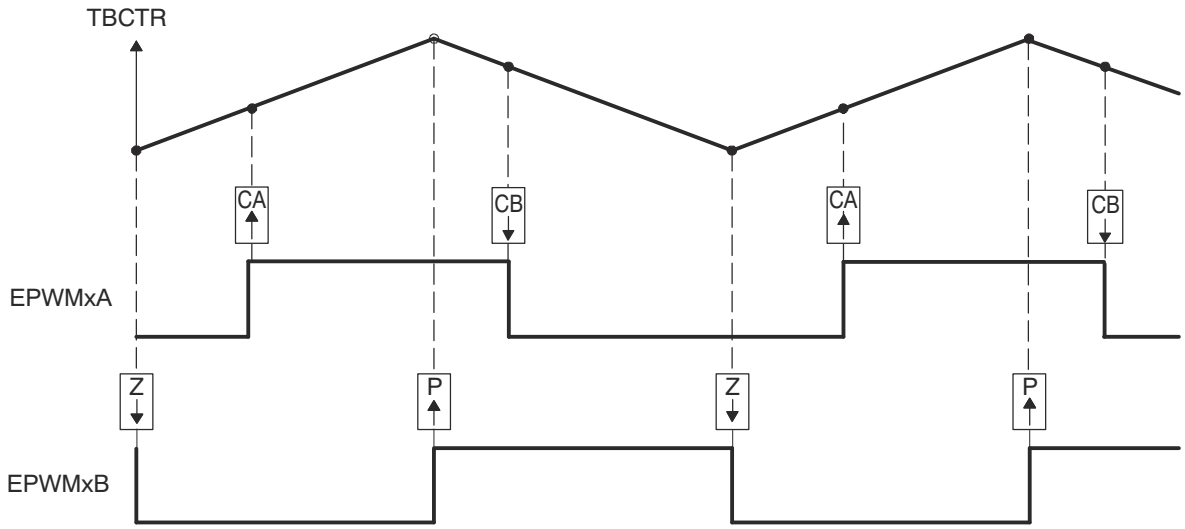
Figure 18-25. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary

Example 18-5 contains a code sample showing initialization and run time for the waveforms in Figure 18-25.

Example 18-5. Code Sample for Figure 18-25

```
// 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 = VCLK4
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
```

18.2.4.9



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 (that is, Set ! Clear and Clear Set).
F. Duty modulation for EPWMxB is fixed at 50% (utilizes spare action resources for EPWMxB)

Figure 18-26. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low

Example 18-6 contains a code sample showing initialization and run time for the waveforms in Figure 18-26.

Example 18-6. Code Sample for Figure 18-26

```
// 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 = VCLK4
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;
```

18.2.5 Dead-Band Generator (DB) Submodule

Figure 18-27 illustrates the dead-band submodule within the ePWM module.

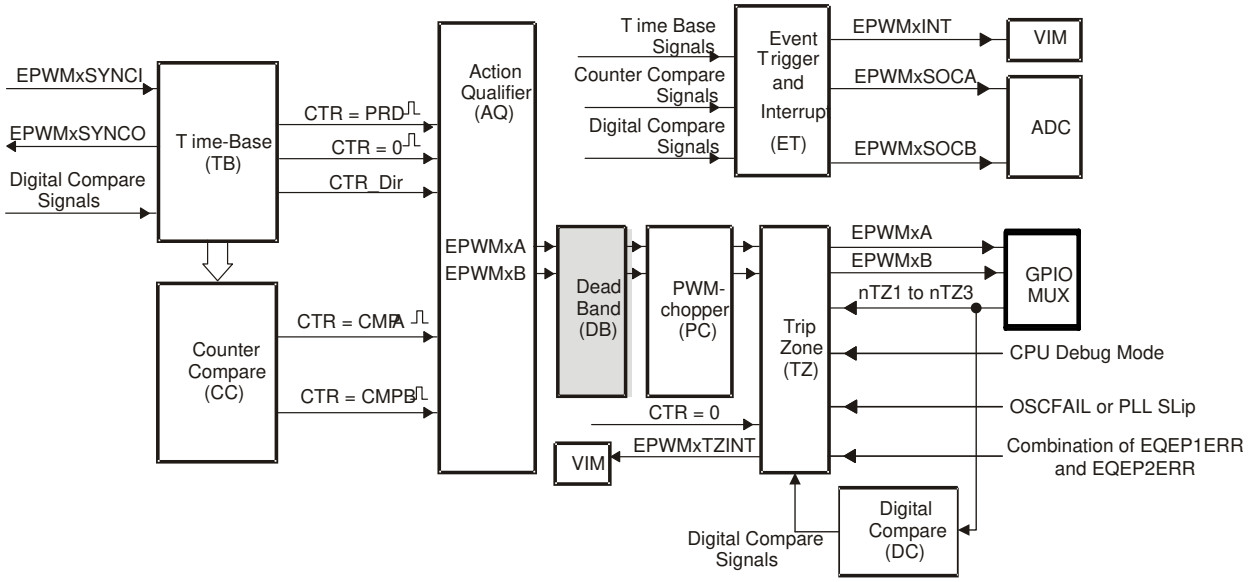


Figure 18-27. Dead_Band Submodule

18.2.5.1 Purpose of the Dead-Band Submodule

Section 18.2.4 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:

ADVANCE INFORMATION

- 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)

18.2.5.2 Controlling and Monitoring the Dead-Band Submodule

The dead-band submodule operation is controlled and monitored via the following registers:

Table 18-13. Dead-Band Generator Submodule Registers

Register Name	Address Offset	Shadowed	Description
DBCTL	0x001C	No	Dead-Band Control Register
DBRED	0x0022	No	Dead-Band Rising Edge Delay Count Register
DBFED	0x0020	No	Dead-Band Falling Edge Delay Count Register

18.2.5.3 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 18-28.

- **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 (that is, 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.

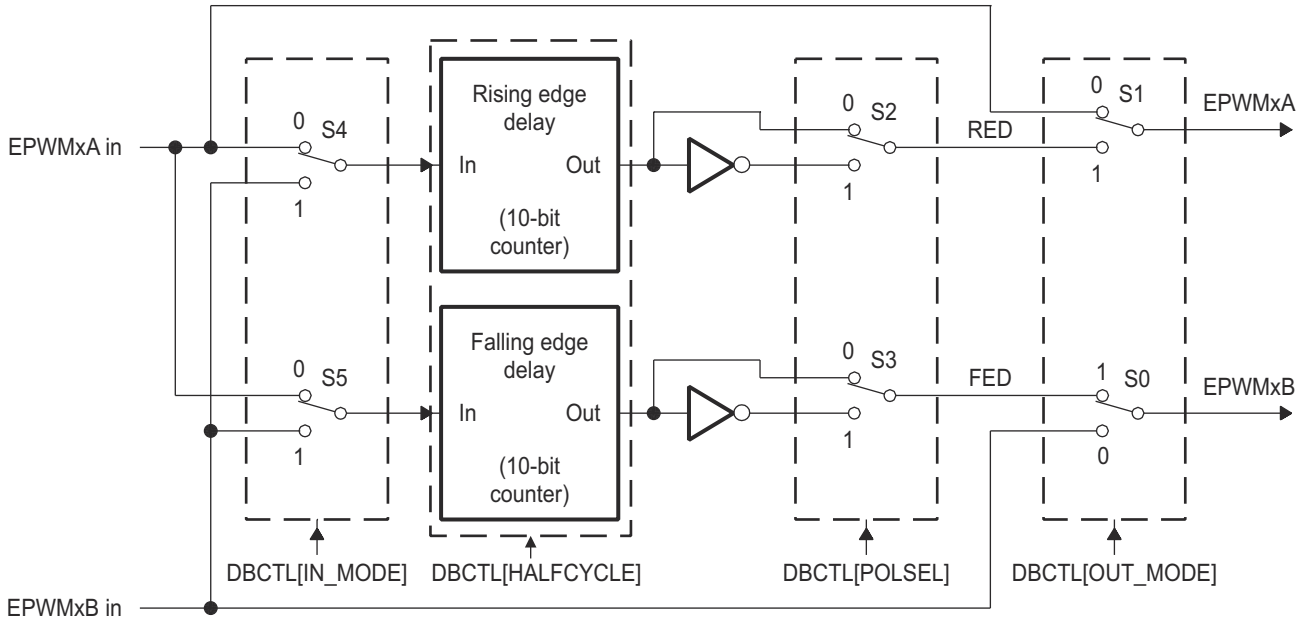


Figure 18-28. Configuration Options for the Dead-Band Submodule

Although all combinations are supported, not all are typical usage modes. Table 18-14 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 18-14 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 18-29. Note that to generate equivalent waveforms to Figure 18-29, 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 18-14 show combinations where either the falling-edge-delay (FED) or rising-edge-delay (RED) blocks are bypassed.

Table 18-14. 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 18-29 shows waveforms for typical cases where 0% < duty < 100%.

ADVANCE INFORMATION

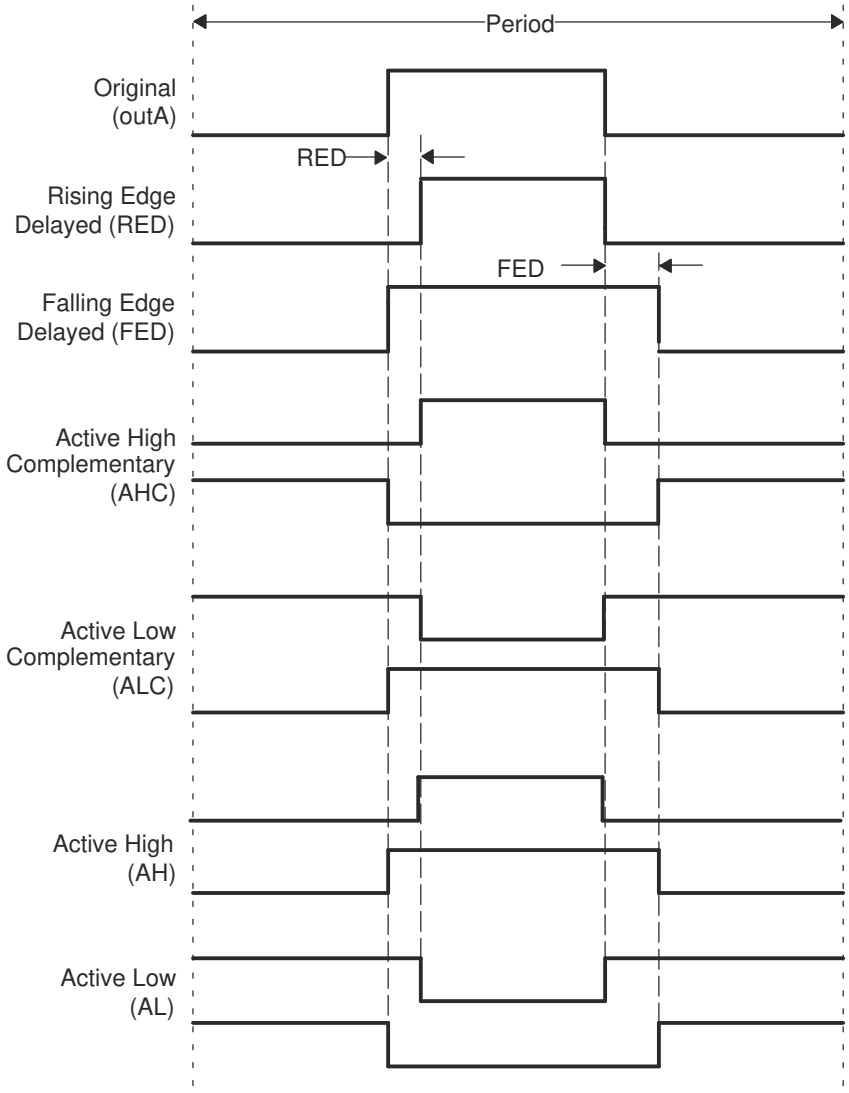


Figure 18-29. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)

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 VCLK4.

For convenience, delay values for various TBCLK options are shown in Table 18-15.

Table 18-15. Dead-Band Delay Values in µS as a Function of DBFED and DBRED

Dead-Band Value	Dead-Band Delay in µS		
DBFED, DBRED	TBCLK = VCLK4/1	TBCLK = VCLK4 /2	TBCLK = VCLK4/4
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	33.33µS	66.67 µS

When half-cycle clocking is enabled, the formula to calculate the falling-edge-delay and rising-edge-delay becomes:

$FED = DBFED \times T_{TBCLK}/2$

$RED = DBRED \times T_{TBCLK}/2$

18.2.6 PWM-Chopper (PC) Submodule

Figure 18-30 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.

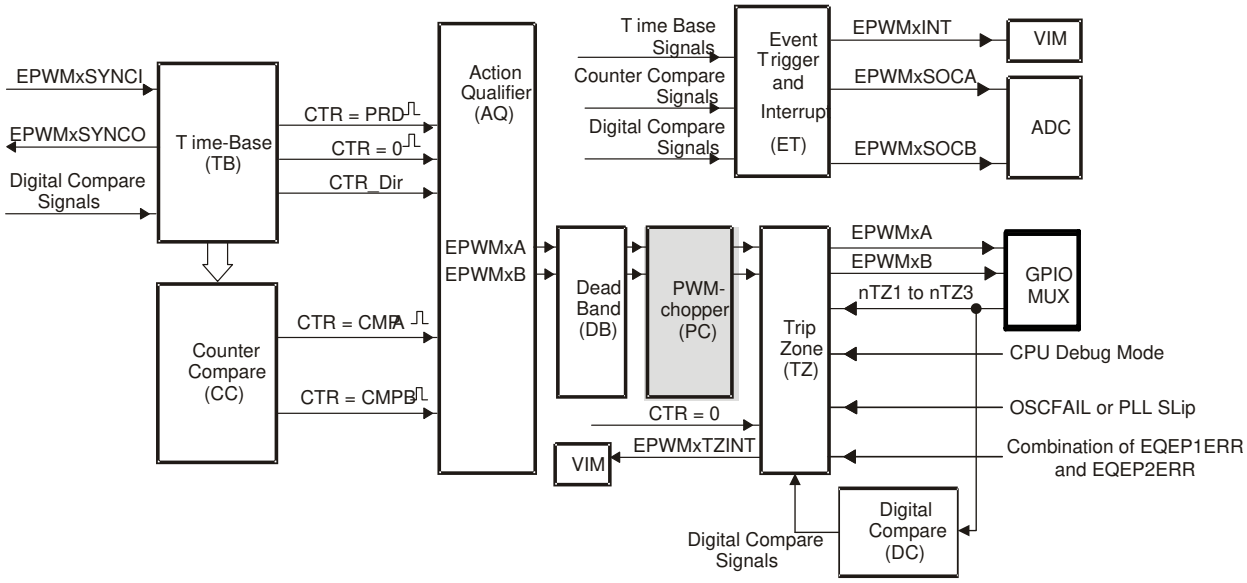


Figure 18-30. PWM-Chopper Submodule

18.2.6.1 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

18.2.6.2 Controlling the PWM-Chopper Submodule

The PWM-chopper submodule operation is controlled via the registers in Table 18-16.

Table 18-16. PWM-Chopper Submodule Registers

Register Name	Address Offset	Shadowed	Description
PCCTL	0x003E	No	PWM-chopper Control Register

18.2.6.3 Operational Highlights for the PWM-Chopper Submodule

Figure 18-31 shows the operational details of the PWM-chopper submodule. The carrier clock is derived from VCLK4. 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.

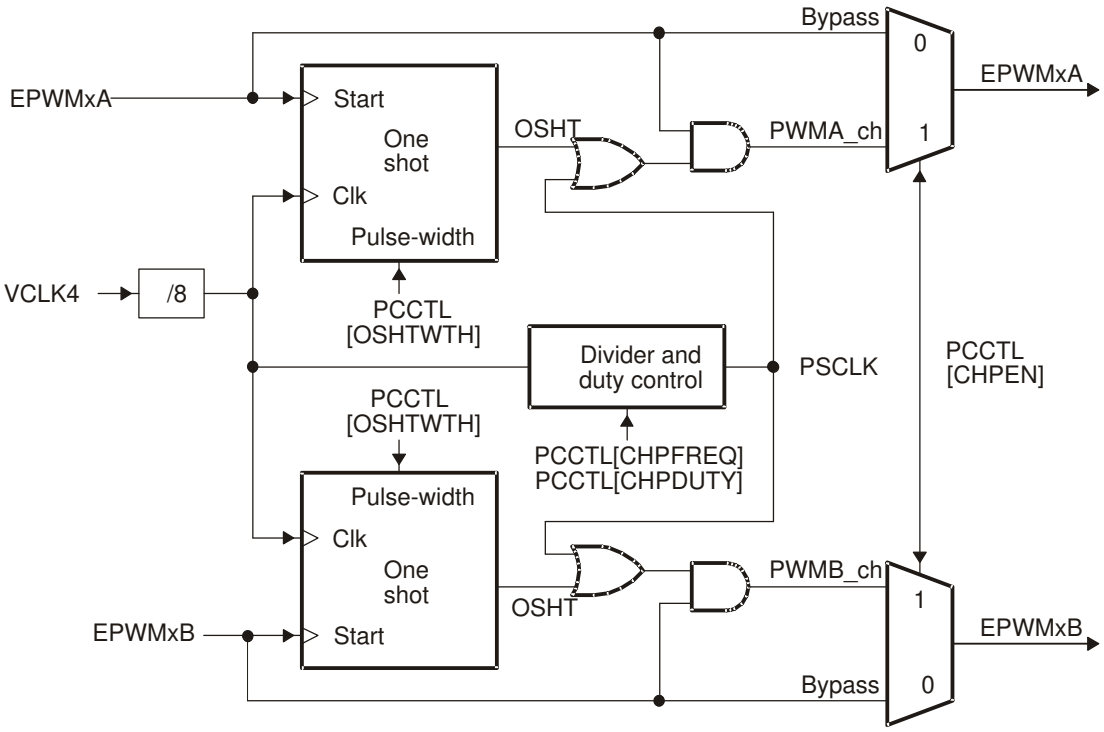


Figure 18-31. PWM-Chopper Submodule Operational Details

18.2.6.4 Waveforms

Figure 18-32 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.

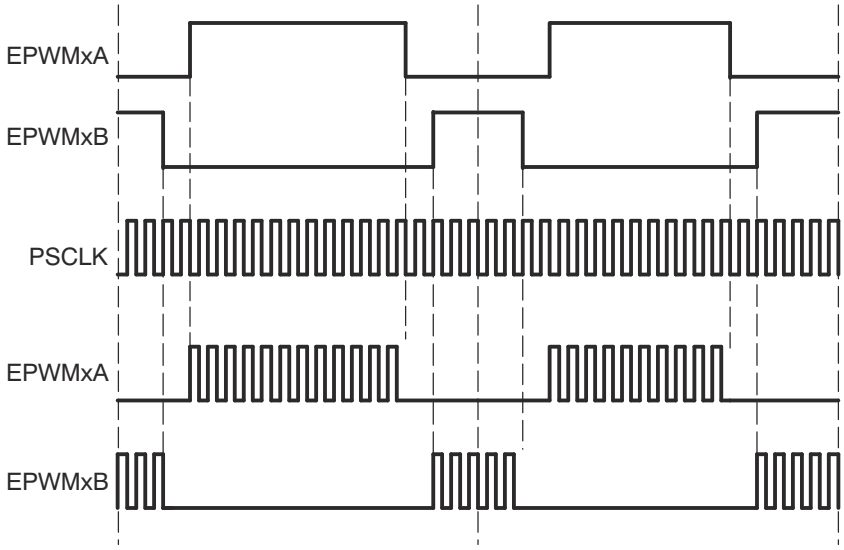


Figure 18-32. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only

18.2.6.4.1 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:

$T_{1stpulse} = T_{VCLK4} \times 8 \times OSHTWTH$

Where T_{VCLK4} is the period of the system clock (VCLK4) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 18-33 shows the first and subsequent sustaining pulses and Table 18-17 gives the possible pulse width values for a VCLK4 = 100 MHz.

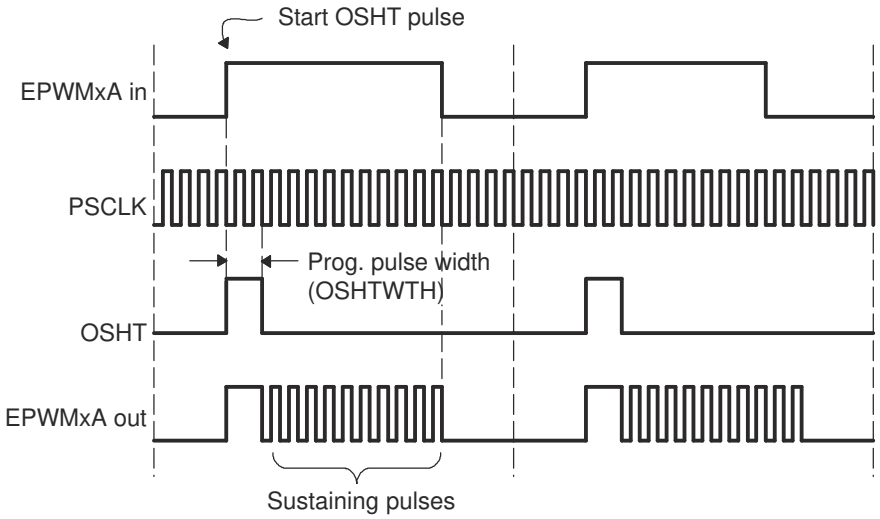


Figure 18-33. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses

Table 18-17. Possible Pulse Width Values for VCLK4 = 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

ADVANCE INFORMATION

18.2.6.4.2 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 18-34 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%.

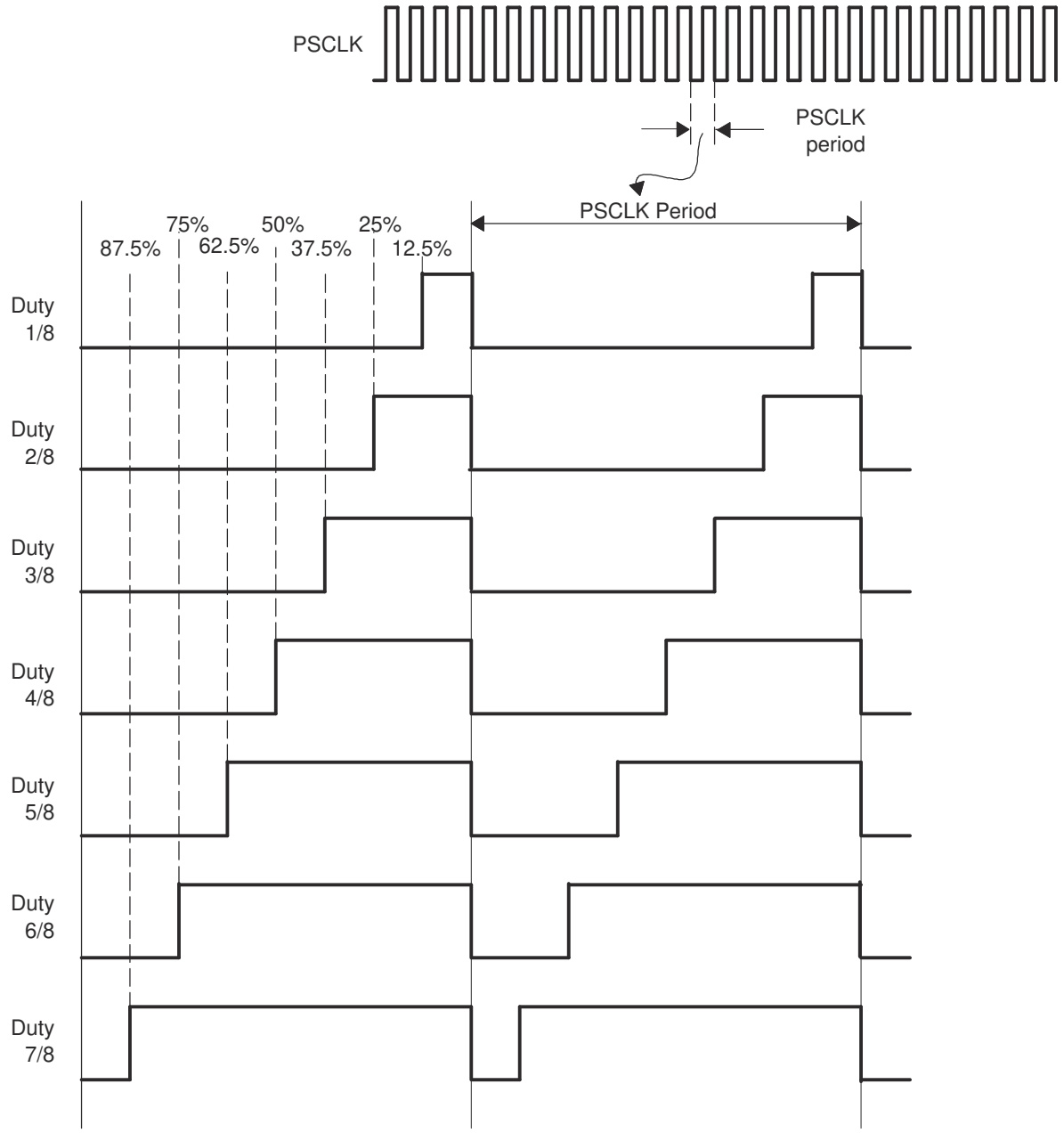


Figure 18-34. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses

18.2.7 Trip-Zone (TZ) Submodule

Figure 18-35 shows how the trip-zone (TZ) submodule fits within the ePWM module.

Each ePWM module is connected to six $\overline{TZn}$ signals ($\overline{TZ1}$ to $\overline{TZ6}$). $\overline{TZ1}$ to $\overline{TZ3}$ are sourced from the GPIO mux. $\overline{TZ4}$ is sourced from a combination of EQEP1ERR and EQEP2ERR signals. $\overline{TZ5}$ is connected to the system oscillator or PLL clock fail logic, and $\overline{TZ6}$ 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.

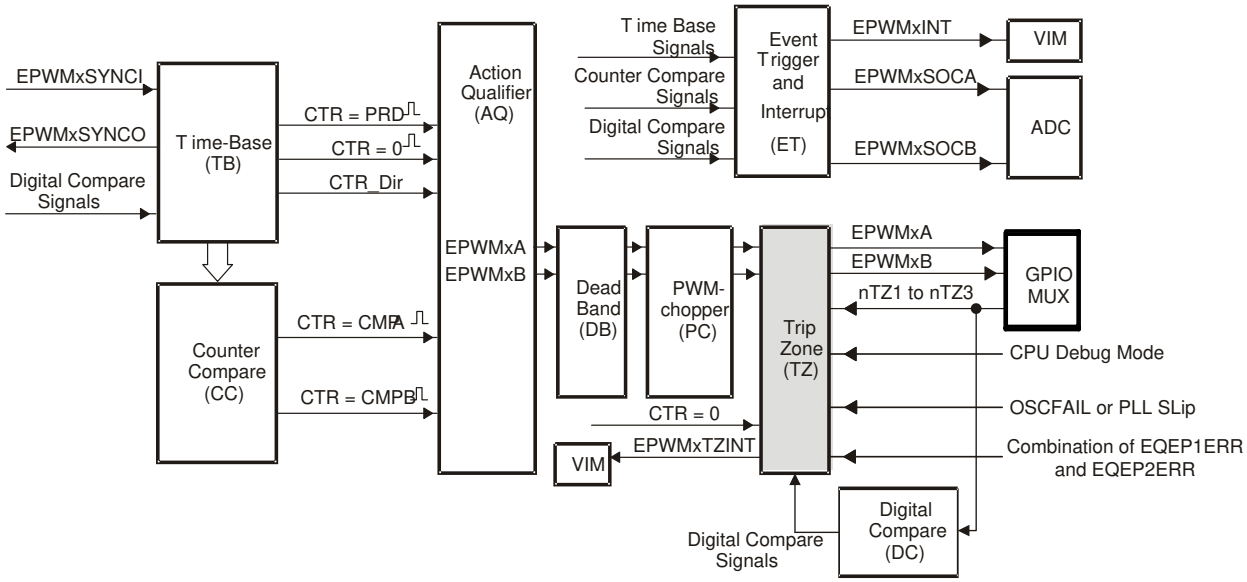


Figure 18-35. Trip-Zone Submodule

18.2.7.1 Purpose of the Trip-Zone Submodule

The key functions of the Trip-Zone submodule are:

- Trip inputs $\overline{TZ1}$ to $\overline{TZ6}$ 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 $\overline{TZ1}$ to $\overline{TZ3}$ 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.

ADVANCE INFORMATION

18.2.7.2 Controlling and Monitoring the Trip-Zone Submodule

The trip-zone submodule operation is controlled and monitored through the following registers:

Table 18-18. Trip-Zone Submodule Registers

Register Name	Address Offset	Shadowed	Description (2)
TZSEL	0x0026	No	Trip-Zone Select Register
TZDCSEL	0x0024	No	Trip-zone Digital Compare Select Register (1)
TZCTL	0x002A	No	Trip-Zone Control Register
TZEINT	0x0028	No	Trip-Zone Enable Interrupt Register
TZFLG	0x002E	No	Trip-Zone Flag Register
TZCLR	0x002C	No	Trip-Zone Clear Register
TZFRC	0x0032	No	Trip-Zone Force Register

(1) This register is discussed in more detail in Section 18.2.9.
(2) All trip-zone registers are writable only in privileged mode.

18.2.7.3 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 (VCLK4) 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 18-19 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 18-19 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 18.2.9.

  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 18-19 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 18-19, can be taken on a trip event.

Table 18-19. 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.

ADVANCE INFORMATION

Example 18-7. 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.

18.2.7.4 Generating Trip Event Interrupts

Figure 18-36 and Figure 18-37 illustrate the trip-zone submodule control and interrupt logic, respectively. DCAEVT1/2 and DCBEVT1/2 signals are described in further detail in Section 18.2.9.

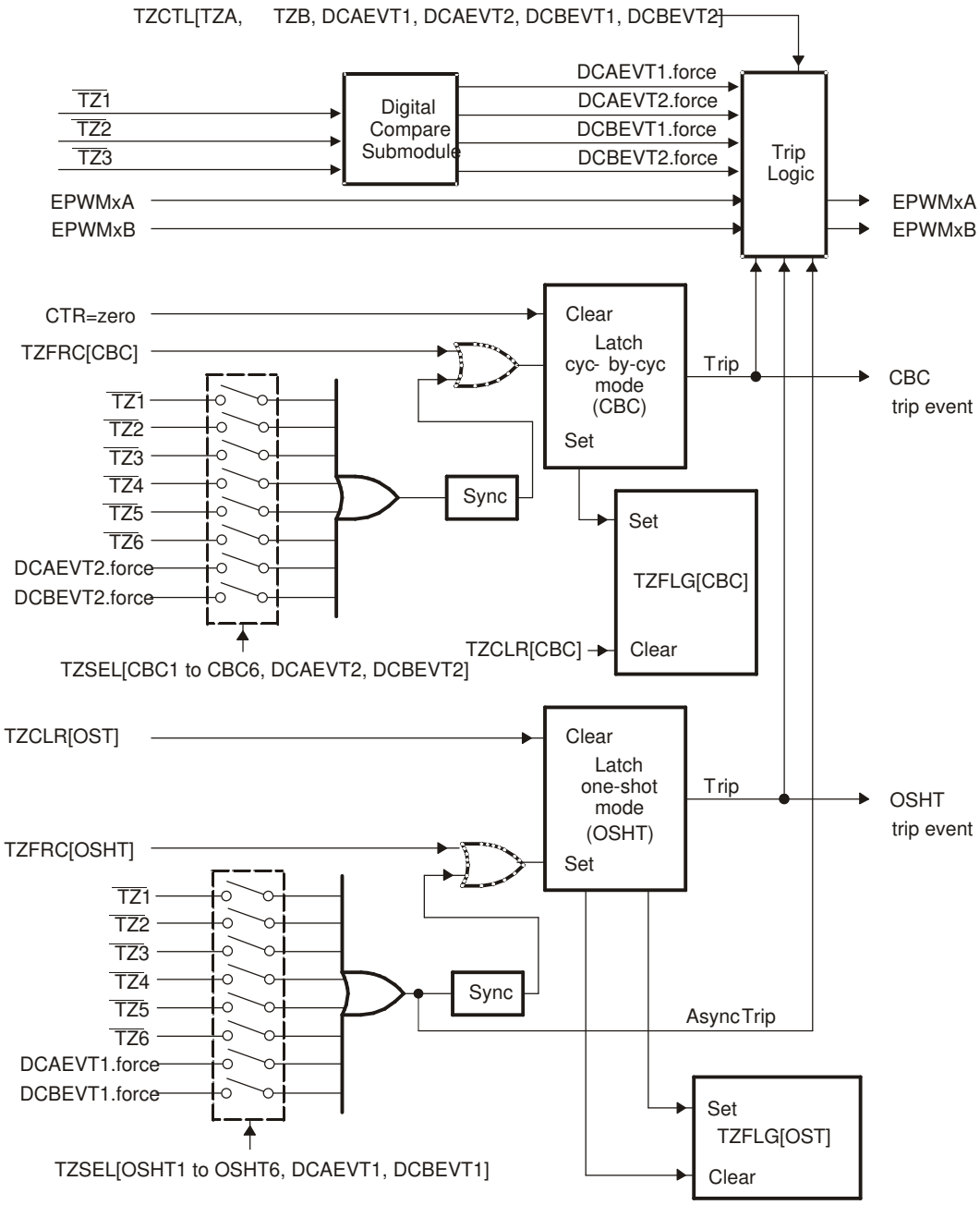


Figure 18-36. Trip-Zone Submodule Mode Control Logic

ADVANCE INFORMATION

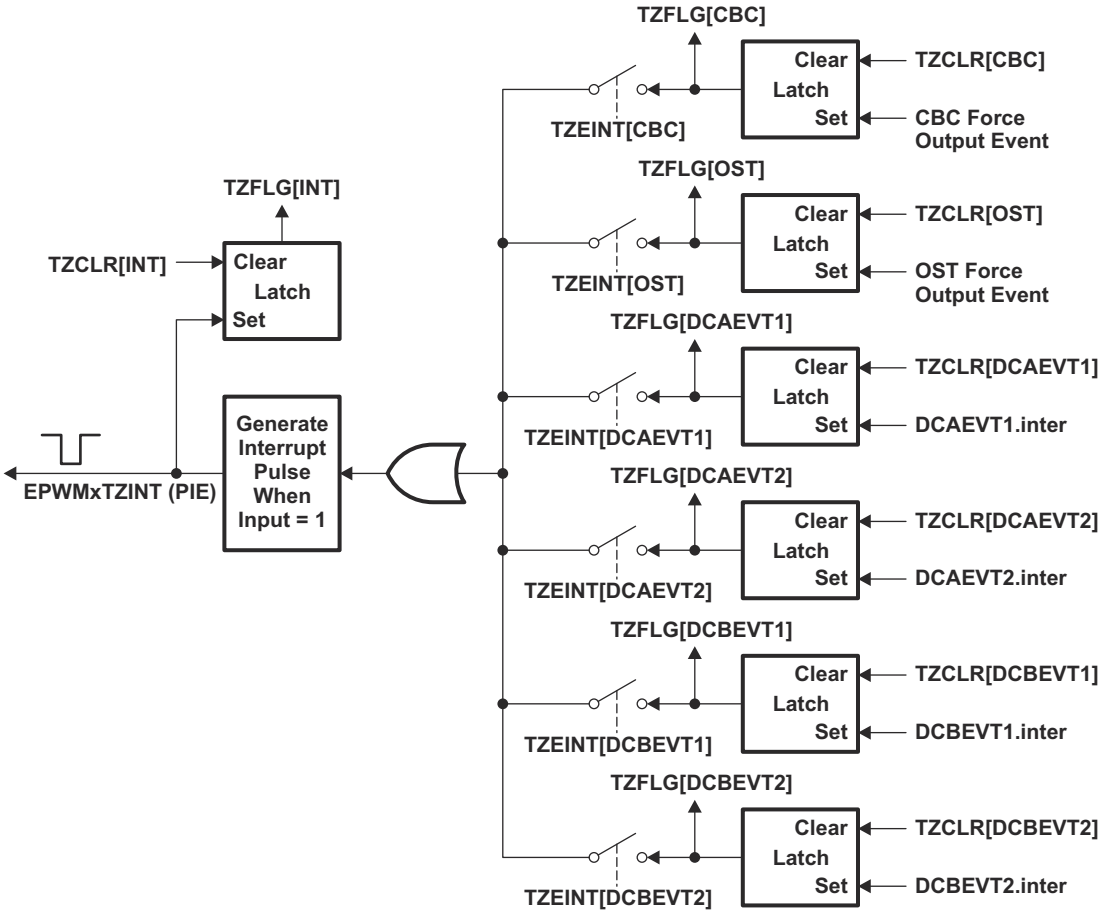


Figure 18-37. Trip-Zone Submodule Interrupt Logic

ADVANCE INFORMATION

18.2.8 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 18-38 illustrates where the event-trigger submodule fits within the ePWM system.

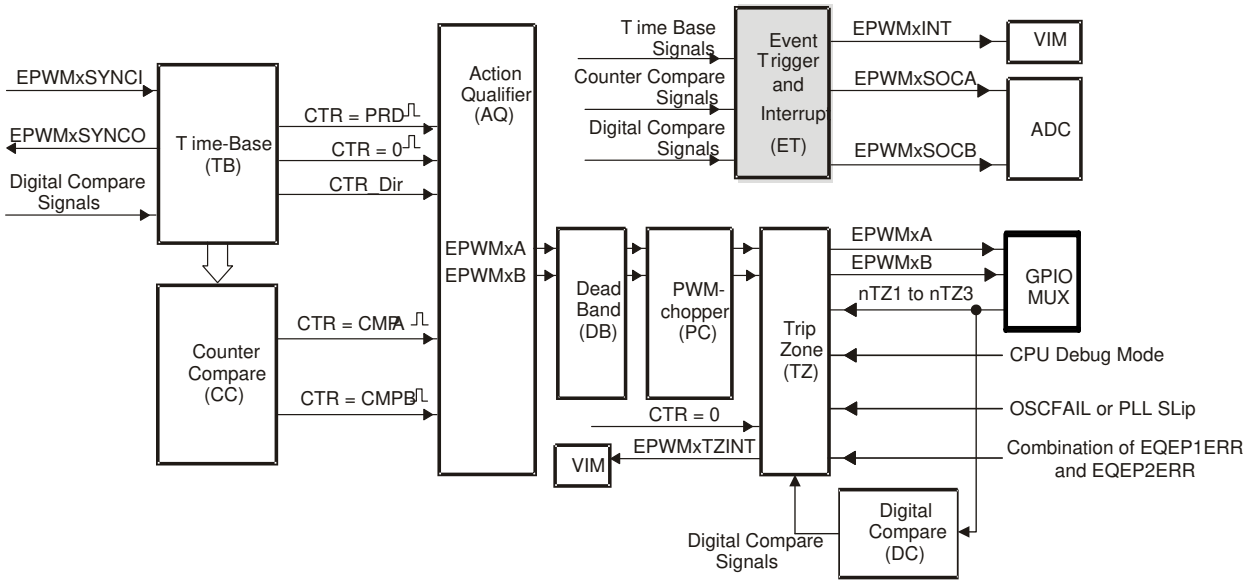


Figure 18-38. Event-Trigger Submodule

18.2.8.1 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 18-39, 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.

The event-trigger submodule monitors various event conditions (the left side inputs to event-trigger submodule shown in Figure 18-40) 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

ADVANCE INFORMATION

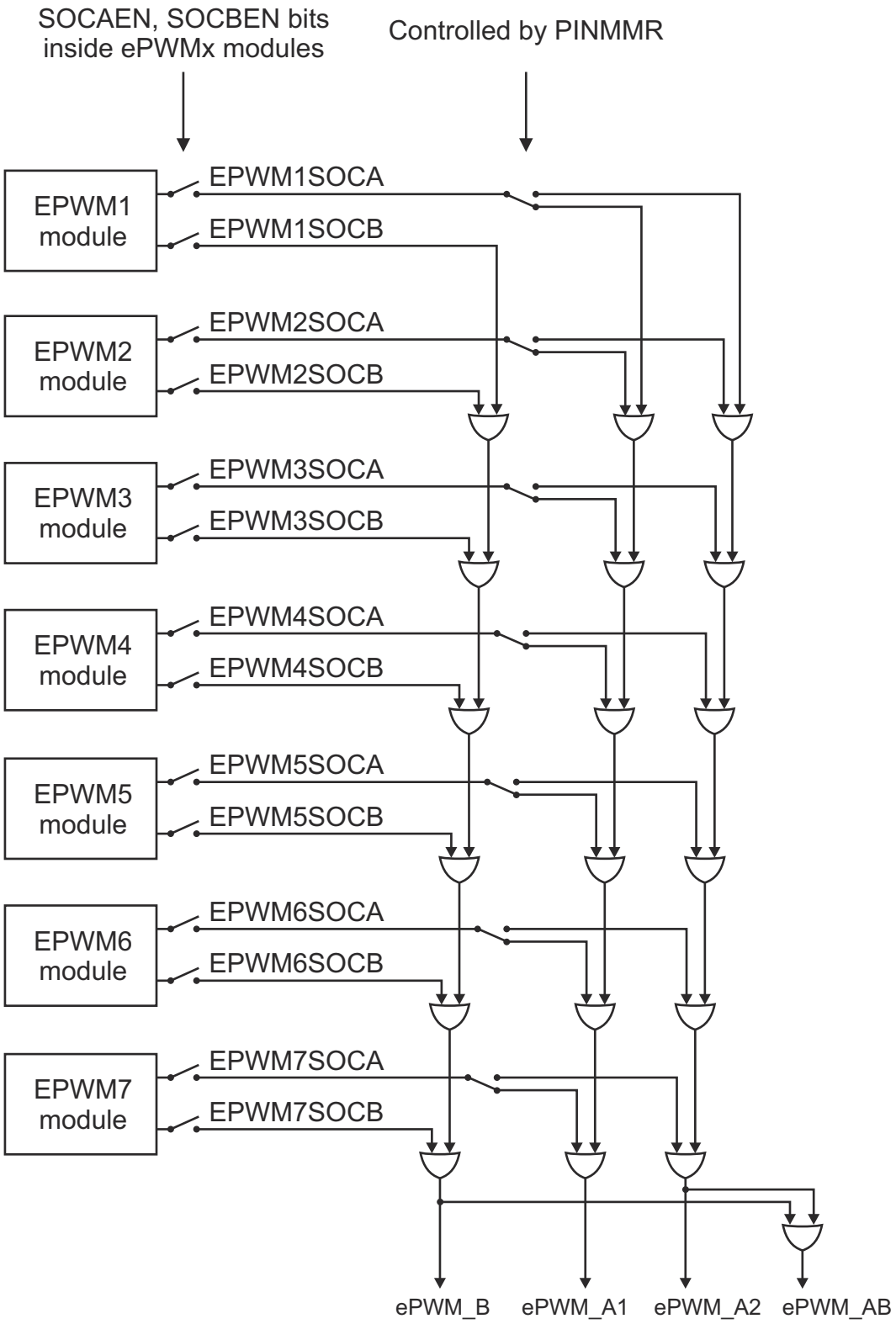


Figure 18-39. Event-Trigger Submodule Inter-Connectivity of ADC Start of Conversion

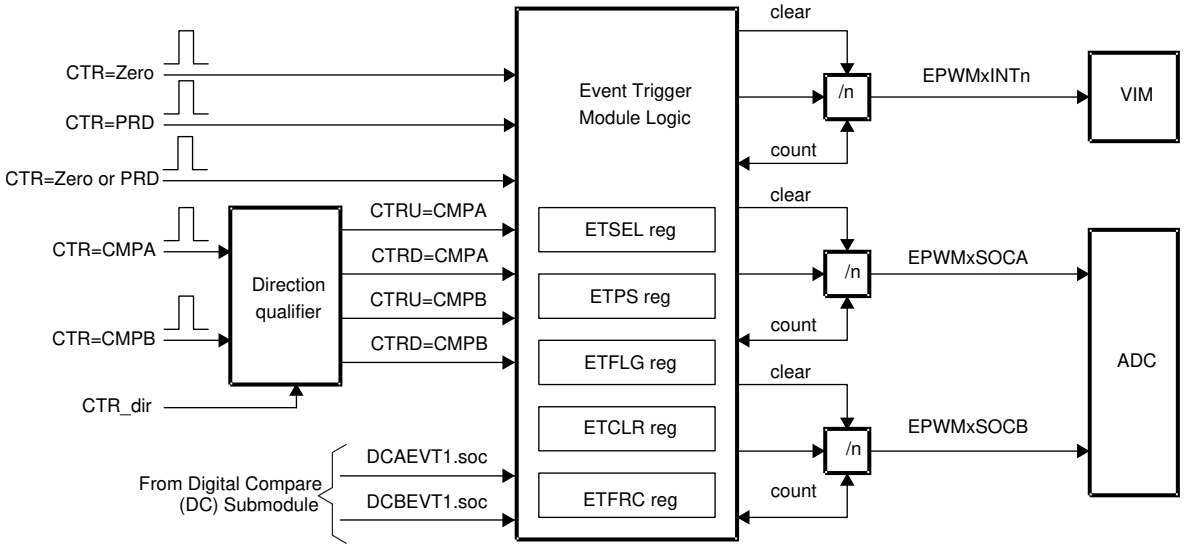


Figure 18-40. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs

The key registers used to configure the event-trigger submodule are shown in Table 18-20.

Table 18-20. Event-Trigger Submodule Registers

Register Name	Address Offset	Shadowed	Description
ETSEL	0x0030	No	Event-trigger Selection Register
ETPS	0x0036	No	Event-trigger Prescale Register
ETFLG	0x0034	No	Event-trigger Flag Register
ETCLR	0x003A	No	Event-trigger Clear Register
ETFRC	0x0038	No	Event-trigger Force 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 18-41, Figure 18-42, and Figure 18-43.

Figure 18-41 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.

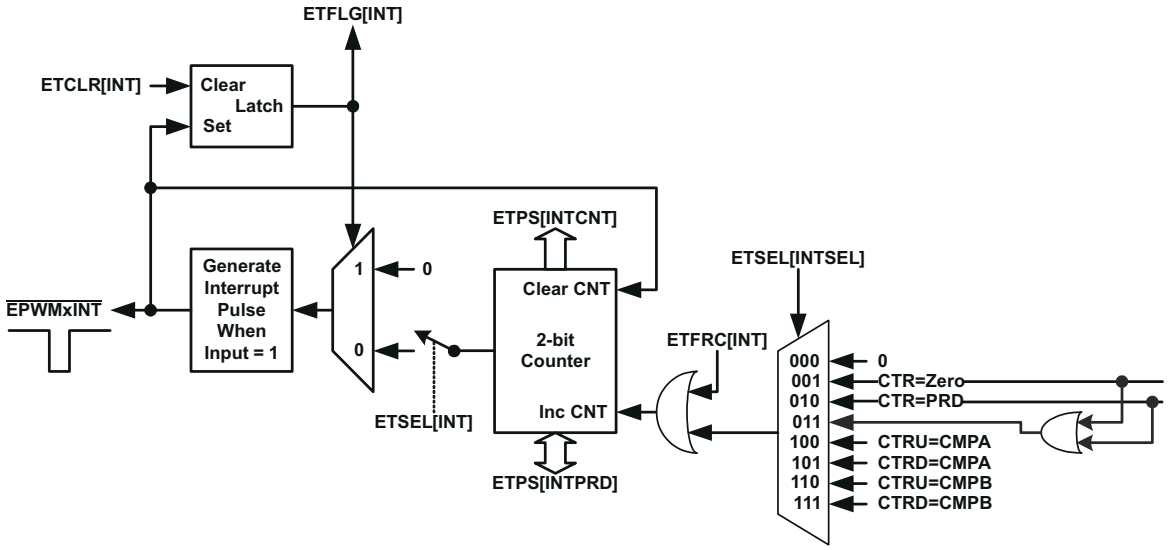
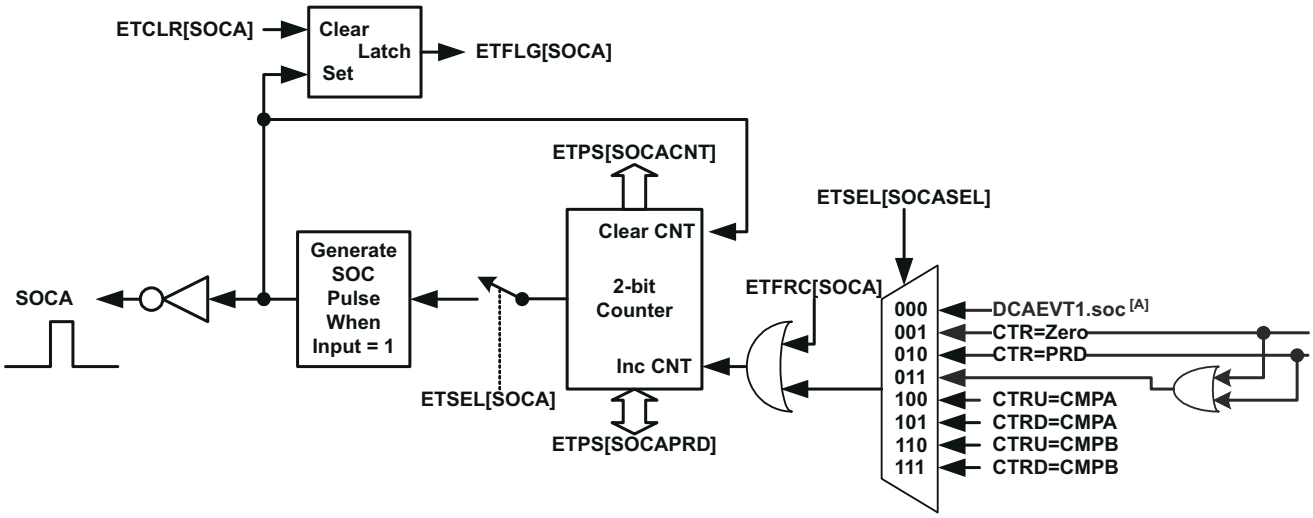


Figure 18-41. Event-Trigger Interrupt Generator

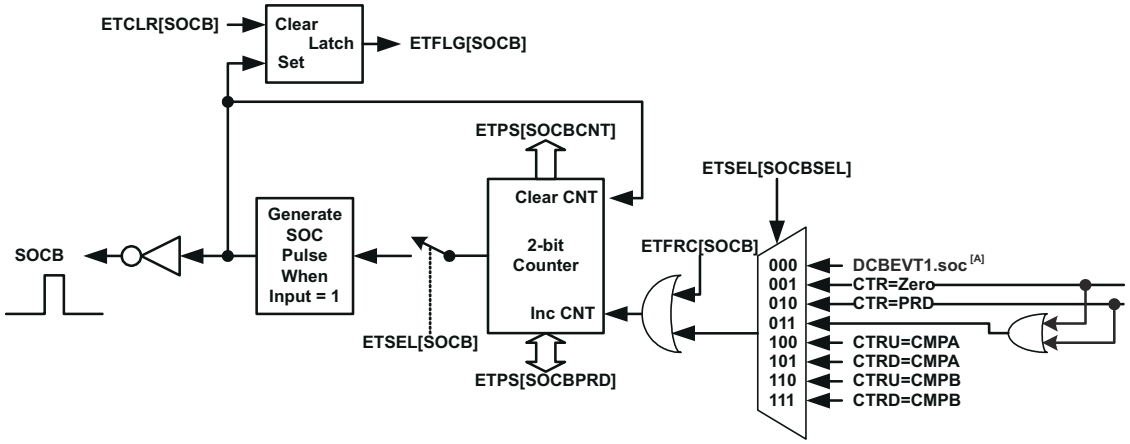
Figure 18-42 shows the operation of the event-trigger's start-of-conversion-A (SOCA) pulse generator. The ETPS[SOCACNT] counter and ETPS[SOCAPRD] period values behave similarly to the interrupt generator except that the pulses are continuously generated. That is, the pulse flag ETFLG[SOCA] is latched 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.



A. The DCAEVT1.soc signals are signals generated by the Digital compare (DC) submodule described later in Section 18.2.9.

Figure 18-42. Event-Trigger SOCA Pulse Generator

Figure 18-43 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.



A. The DCBEVT1.soc signals are signals generated by the Digital compare (DC) submodule described later in Section 18.2.9.

Figure 18-43. Event-Trigger SOCB Pulse Generator

ADVANCE INFORMATION

18.2.9 Digital Compare (DC) Submodule

Figure 18-44 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 that 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.

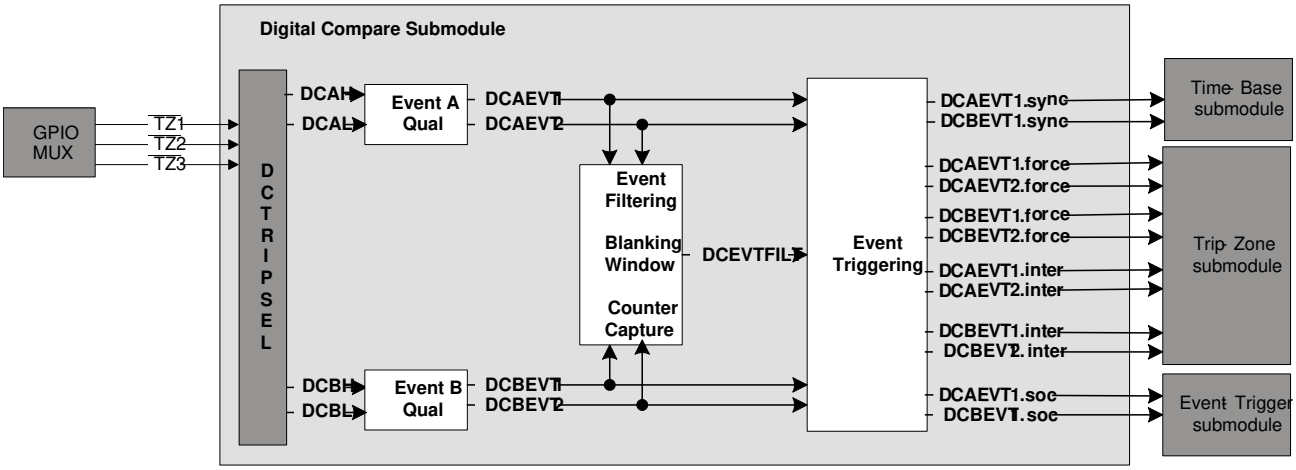


Figure 18-44. Digital-Compare Submodule High-Level Block Diagram

18.2.9.1 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.

18.2.9.2 Controlling and Monitoring the Digital Compare Submodule

The digital compare submodule operation is controlled and monitored through the following registers:

Table 18-21. Digital Compare Submodule Registers

Register Name	Address Offset	Shadowed	Description
TZDCSEL(1) (2)	0x0024	No	Trip Zone Digital Compare Select Register
DCTRIPSEL(1)	0x0062	No	Digital Compare Trip Select Register
DCACTL(1)	0x0060	No	Digital Compare A Control Register
DCBCTL(1)	0x0066	No	Digital Compare B Control Register
DCFCTL(1)	0x0064	No	Digital Compare Filter Control Register
DCCAPCTL(1)	0x006A	No	Digital Compare Capture Control Register
DCFOFFSET	0x0068	Writes	Digital Compare Filter Offset Register
DCFOFFSETCNT	0x006E	No	Digital Compare Filter Offset Counter Register
DCFWINDOW	0x006C	No	Digital Compare Filter Window Register
DCFWINDOWCNT	0x0072	No	Digital Compare Filter Window Counter Register
DCCAP	0x0070	Yes	Digital Compare Counter Capture 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.

18.2.9.3 Operation Highlights of the Digital Compare Submodule

The following sections describe the operational highlights and configuration options for the digital compare submodule.

18.2.9.3.1 Digital Compare Events

As illustrated in Figure 18-44, 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.

ADVANCE INFORMATION

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 18.2.9.3.2. 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 18-45 and Figure 18-46 show how the DCAEVT1, DCAEVT2, or DCEVTFLT signals are processed to generate the digital compare A event force, interrupt, soc and sync signals.

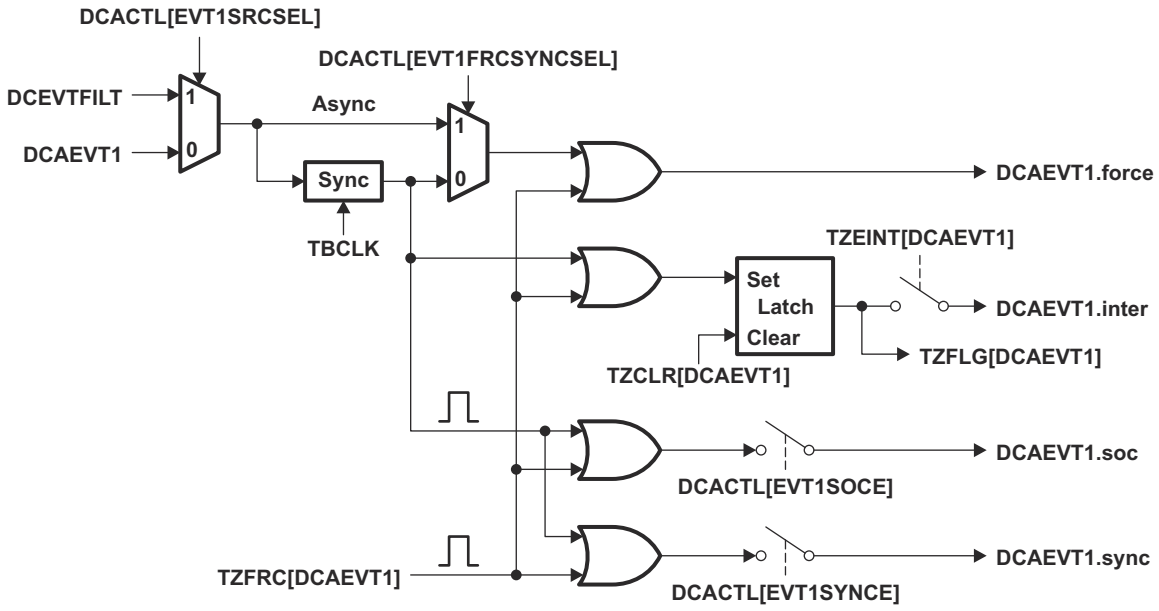


Figure 18-45. DCAEVT1 Event Triggering

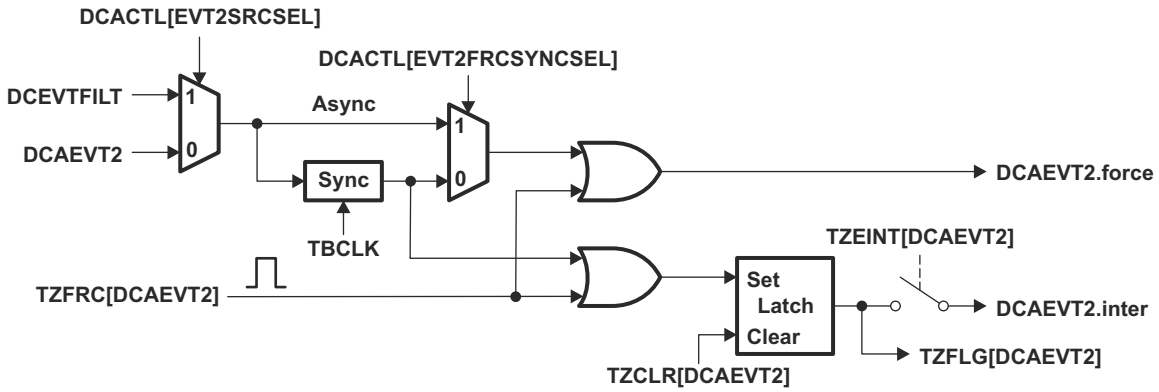


Figure 18-46. DCAEVT2 Event Triggering

Figure 18-47 and Figure 18-48 show how the DCBEVT1, DCBEVT2, or DCEVTFLT signals are processed to generate the digital compare B event force, interrupt, soc, and sync signals.

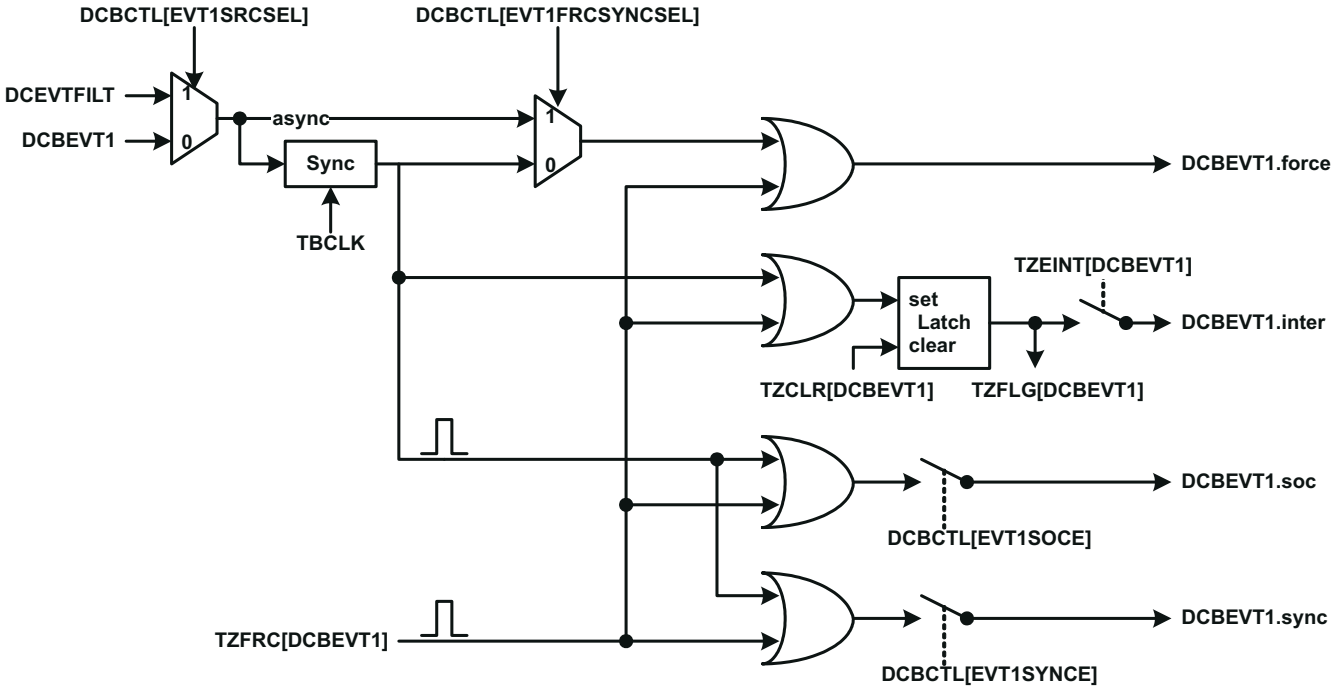


Figure 18-47. DCBEVT1 Event Triggering

ADVANCE INFORMATION

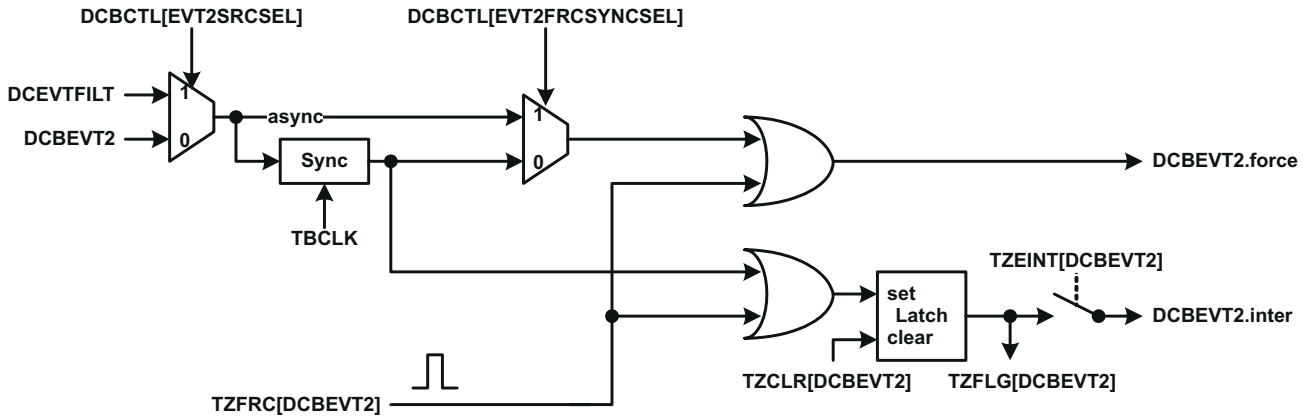


Figure 18-48. DCBEVT2 Event Triggering

ADVANCE INFORMATION

18.2.9.3.2 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 18-49 shows the details of the event filtering logic.

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 18-50 illustrates 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.

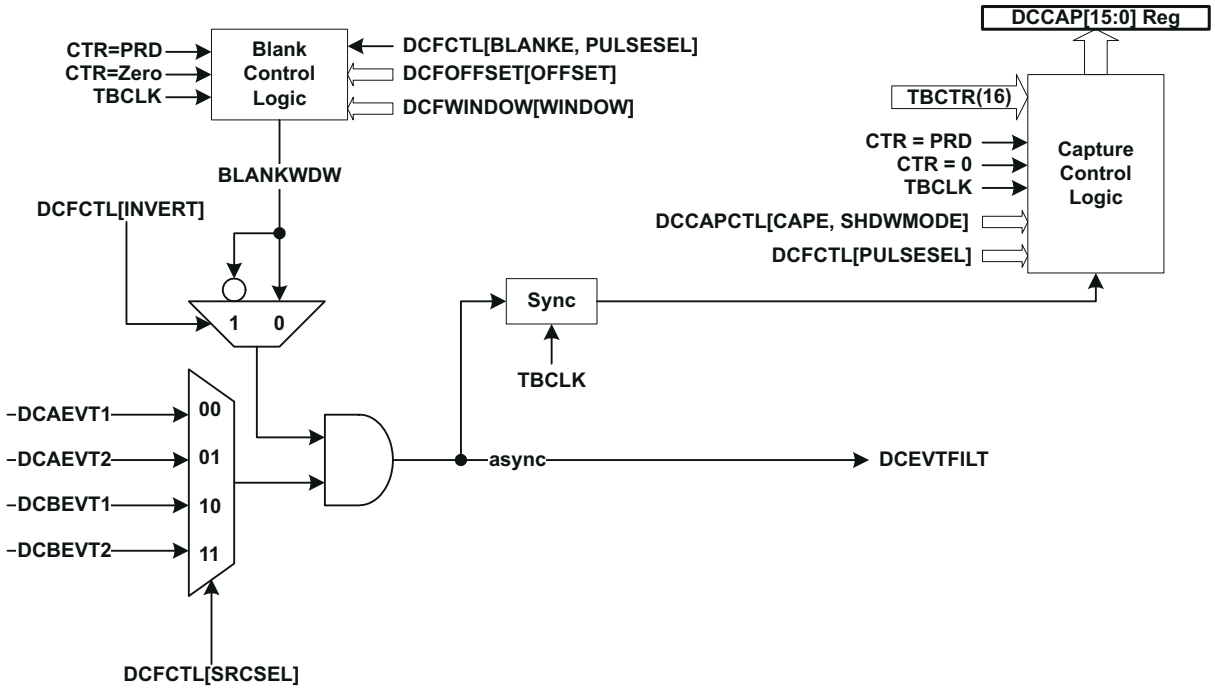


Figure 18-49. Event Filtering

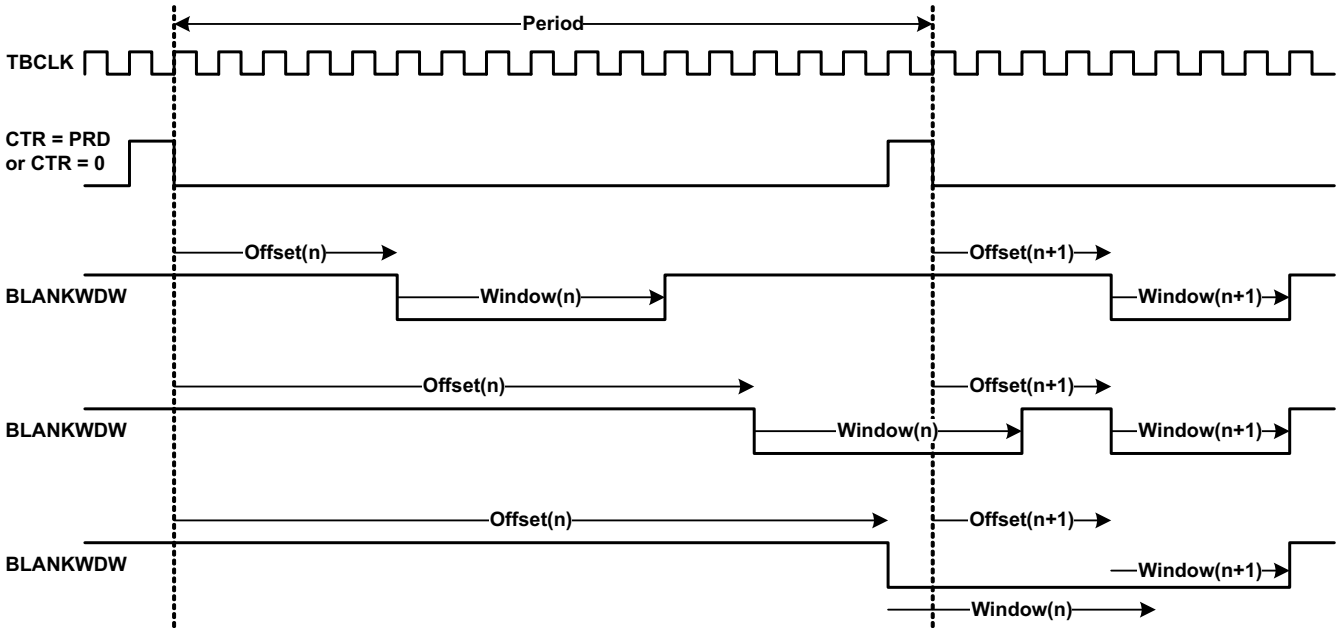


Figure 18-50. Blanking Window Timing Diagram

ADVANCE INFORMATION

18.2.10 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

18.3 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.

18.3.1 Overview of Multiple Modules

Previously in this chapter, 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 18-51. 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.

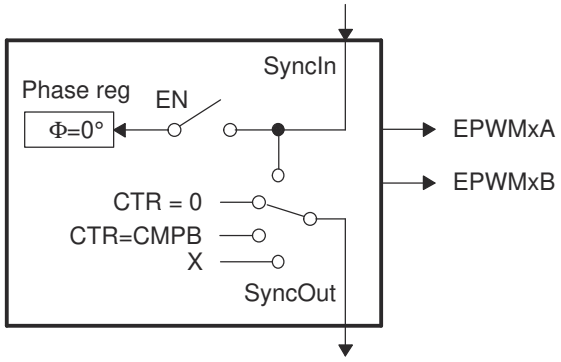


Figure 18-51. Simplified ePWM Module

18.3.2 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
  - Controller mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Controller 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
  - Controller mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Controller 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, that is, via the enable switch. Although various combinations are possible, the two most common—controller module and target module modes—are shown in Figure 18-52.

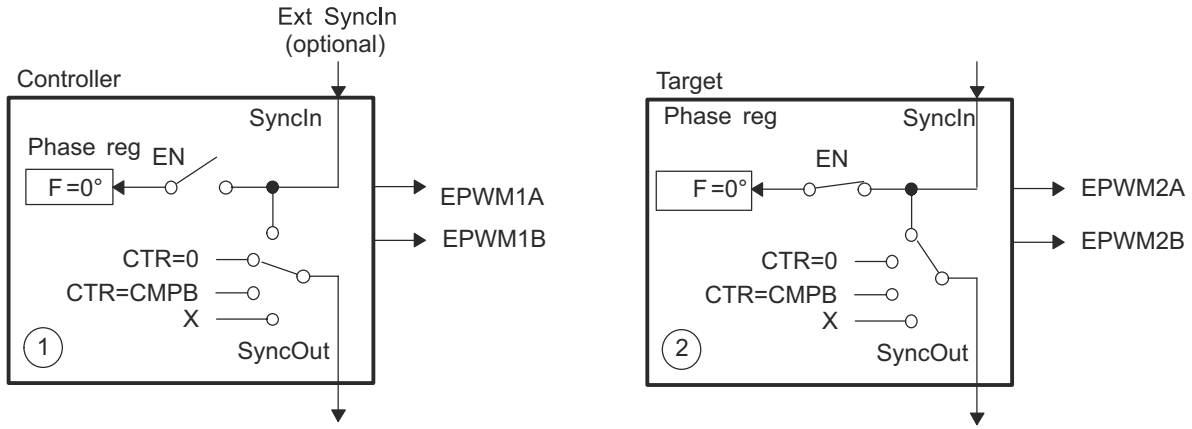
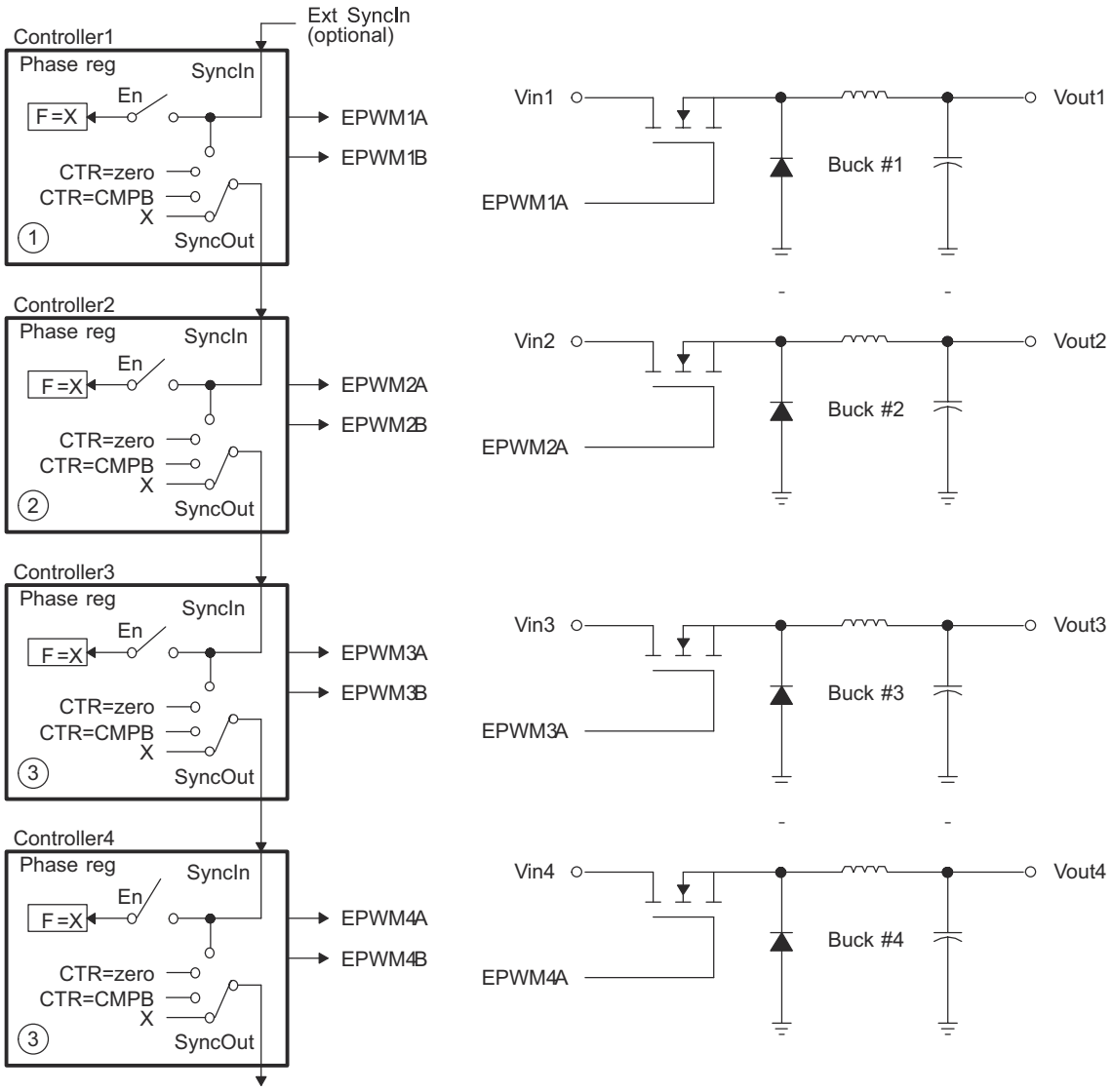


Figure 18-52. EPWM1 Configured as a Typical Controller, EPWM2 Configured as a Target

ADVANCE INFORMATION

18.3.3 Controlling Multiple Buck Converters With Independent Frequencies

One of the simplest power converter topologies is the buck. A single ePWM module configured as a controller 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 18-53 shows four buck stages, each running at independent frequencies. In this case, all four ePWM modules are configured as Controllers and no synchronization is used. Figure 18-54 shows the waveforms generated by the setup shown in Figure 18-53; note that only three waveforms are shown, although there are four stages.



A. Θ = X indicates value in phase register is a "don't care"

Figure 18-53. Control of Four Buck Stages. Here $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$

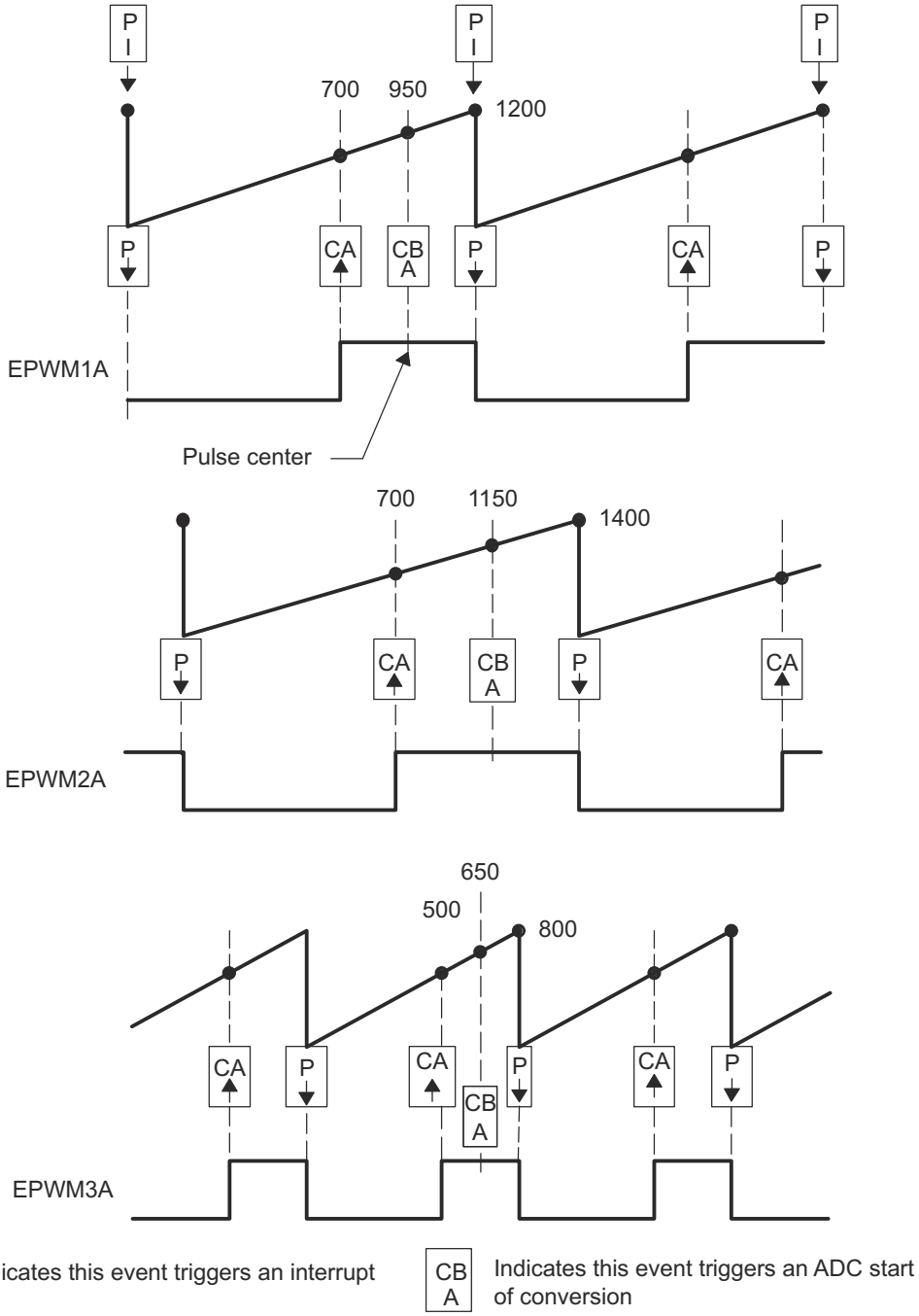


Figure 18-54. Buck Waveforms for Figure 18-53 (Note: Only three bucks shown here)

Example 18-8. Configuration for Example in Figure 18-54

```
//=====================================================================
// (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
```

18.3.4 Controlling Multiple Buck Converters With Same Frequencies

If synchronization is a requirement, ePWM module 2 can be configured as a target and can operate at integer multiple (N) frequencies of module 1. The sync signal from controller to target ensures these modules remain locked. Figure 18-55 shows such a configuration; Figure 18-56 shows the waveforms generated by the configuration.

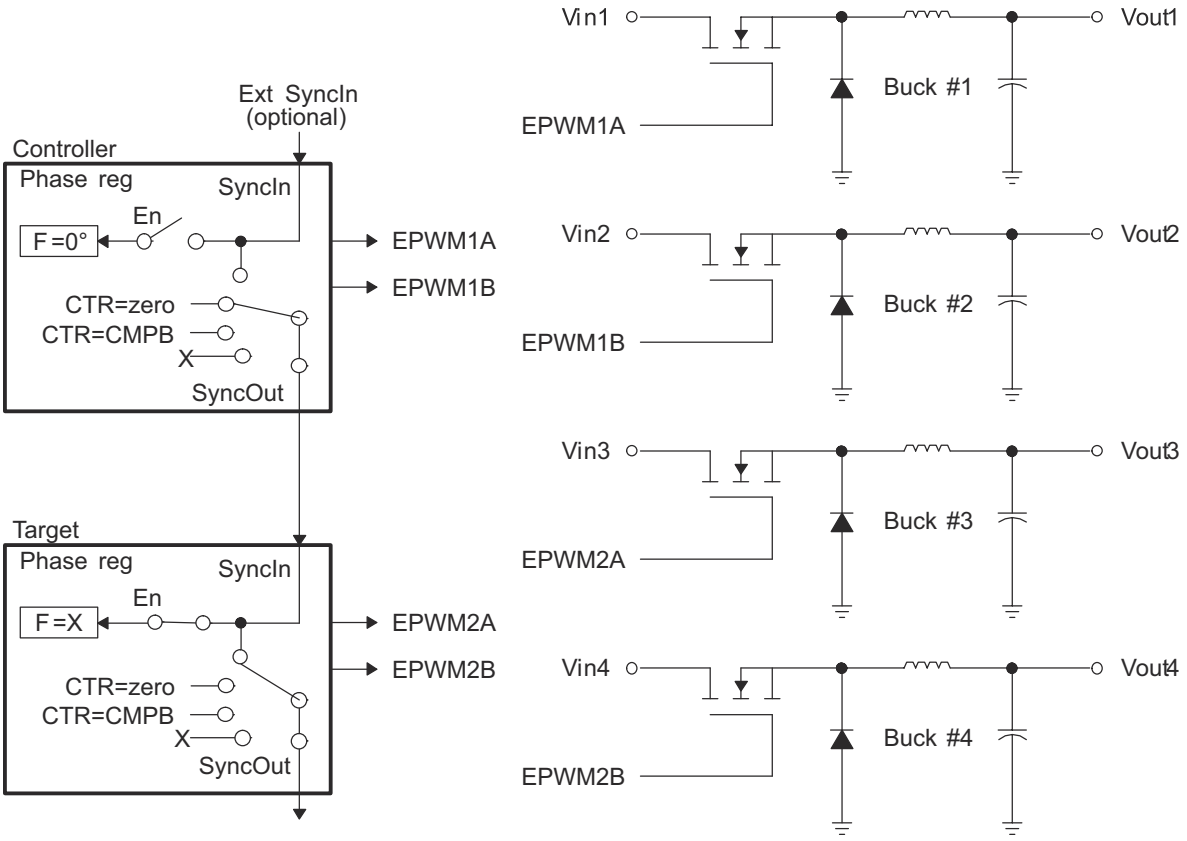


Figure 18-55. Control of Four Buck Stages. (Note: F_{PWM2} = N x F_{PWM1})

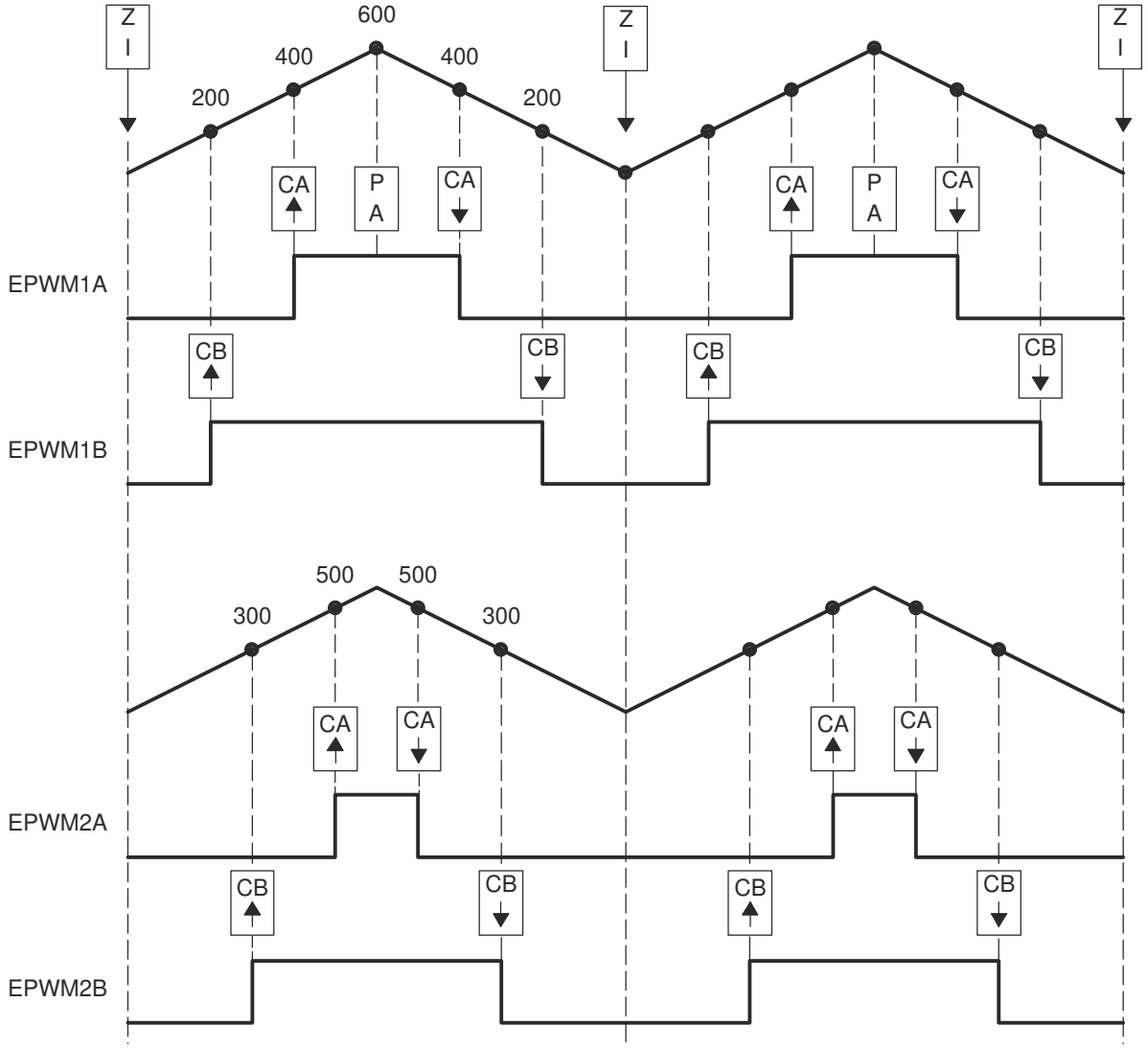


Figure 18-56. Buck Waveforms for Figure 18-55 (Note: $F_{PWM2} = F_{PWM1)}$)

ADVANCE INFORMATION

Example 18-9. Code Snippet for Configuration in Figure 18-55

```
//=======================
// 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;         // Controller 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;          // Target 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
```

18.3.5 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 18-57 shows control of two synchronized Half-H bridge stages where stage 2 can operate at integer multiple (N) frequencies of stage 1. Figure 18-58 shows the waveforms generated by the configuration shown in Figure 18-57.

Module 2 (target) 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 controller module 1.

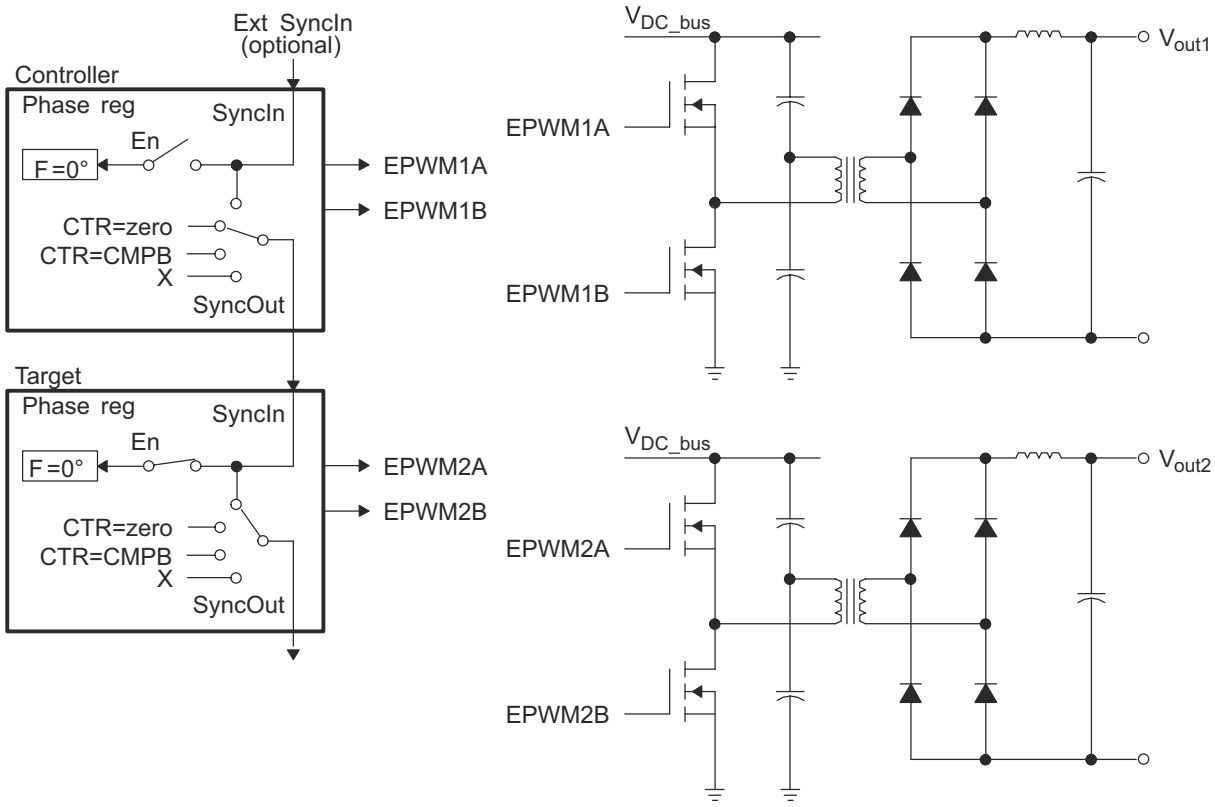


Figure 18-57. Control of Two Half-H Bridge Stages (F_{PWM2} = N x F_{PWM1})

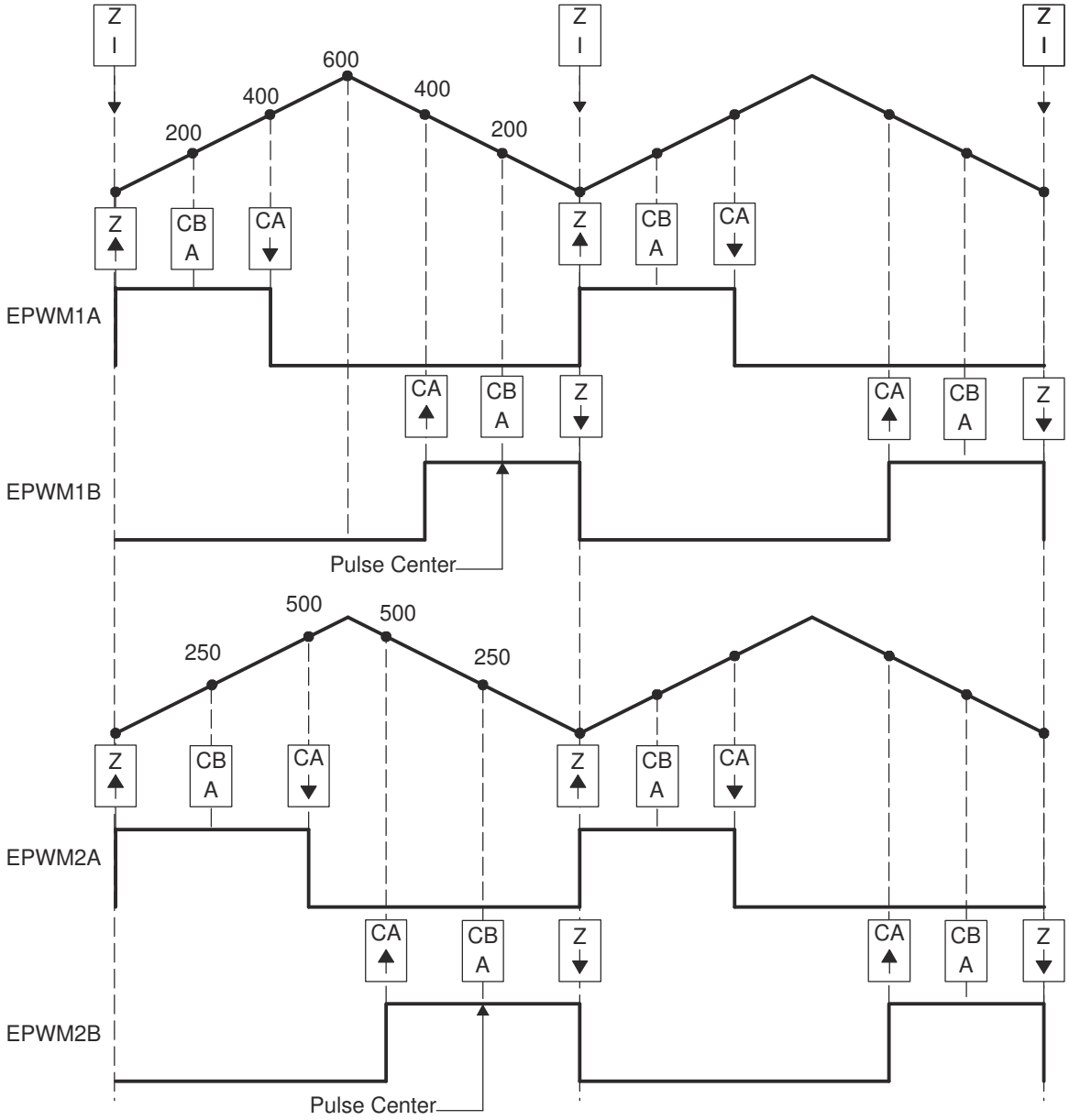


Figure 18-58. Half-H Bridge Waveforms for Figure 18-57 (Note: Here $F_{PWM2} = F_{PWM1}$)

Example 18-10. Code Snippet for Configuration in Figure 18-57

```
//=====================================================================
// 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;          // Controller 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;           // Target 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
```

18.3.6 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 controller + two targets configuration can easily address this requirement. Figure 18-59 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 controllers as in Figure 18-59), or both inverters can be synchronized by using one controller (module 1) and five targets. 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).

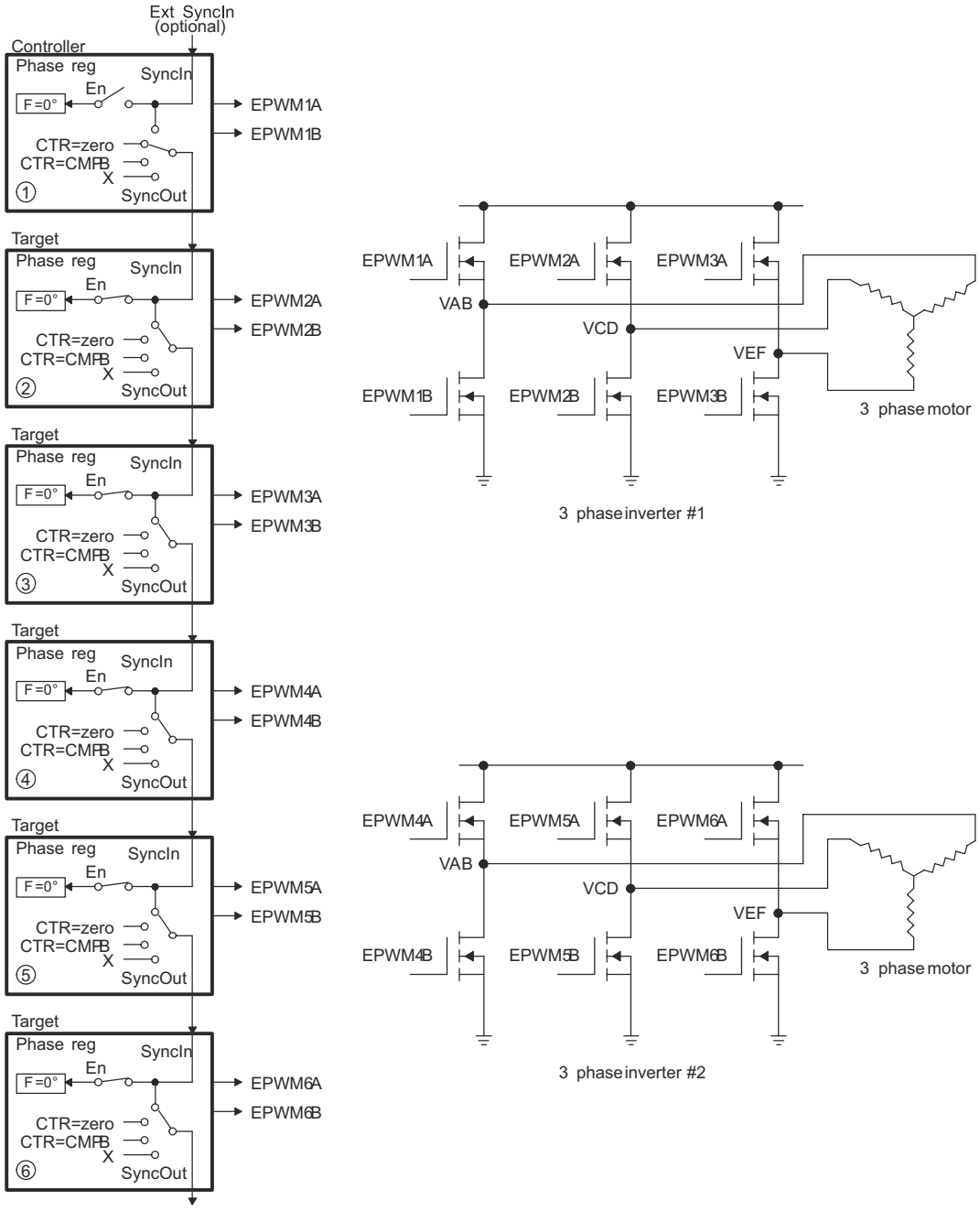


Figure 18-59. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control

ADVANCE INFORMATION

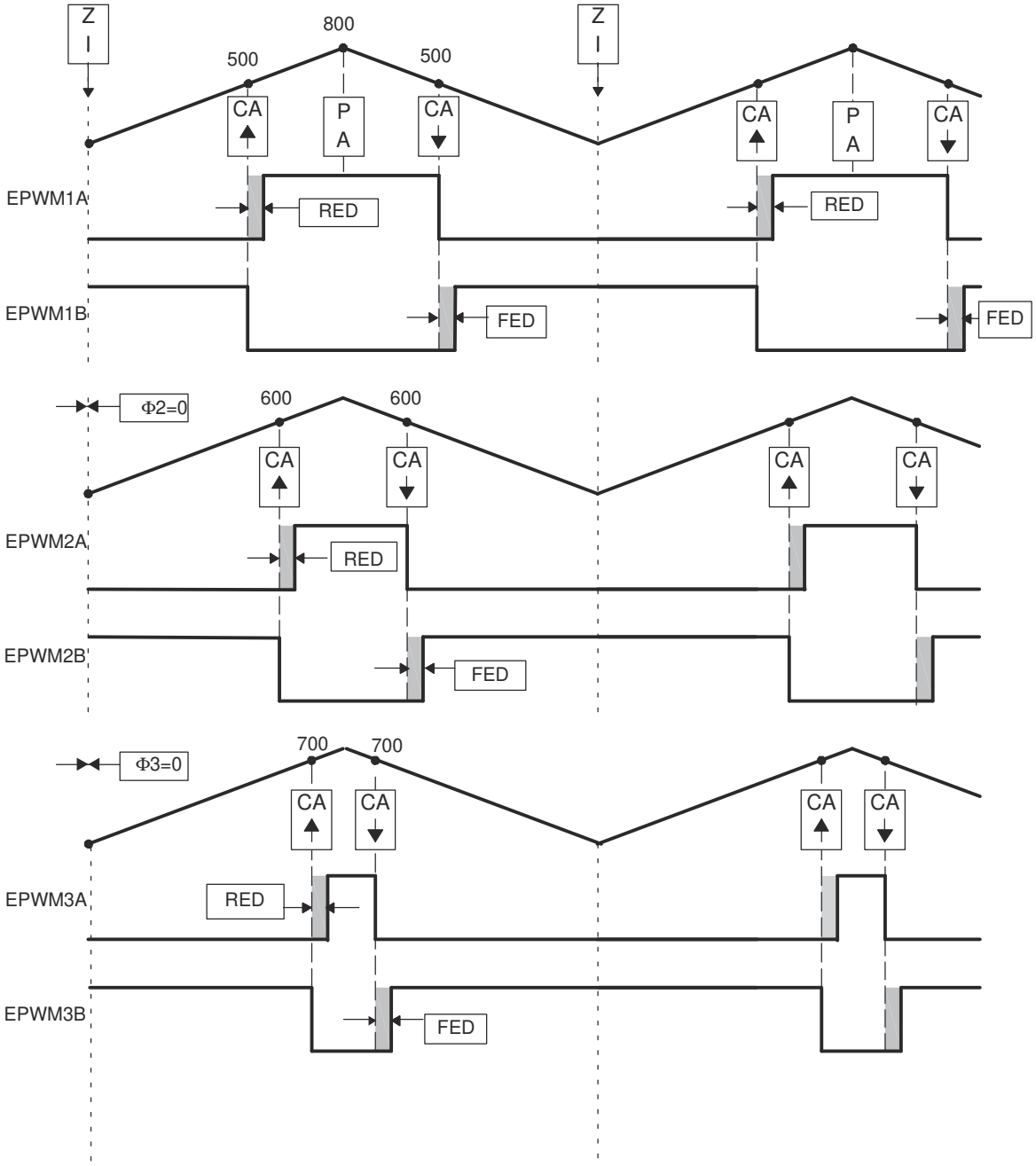


Figure 18-60. 3-Phase Inverter Waveforms for Figure 18-59 (Only One Inverter Shown)

18.3.7 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 18-61 shows a controller and target module with a phase relationship of 120°, that is, the target leads the controller.

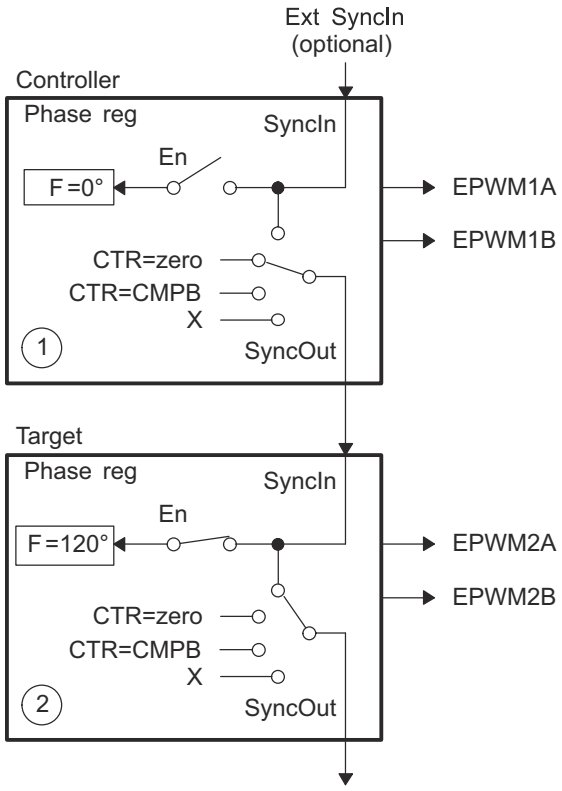


Figure 18-61. Configuring Two PWM Modules for Phase Control

Figure 18-62 shows the associated timing waveforms for this configuration. Here, TBPRD = 600 for both controller and target. For the target, TBPHS = 200 (that is, 200/600 X 360° = 120°). Whenever the controller generates a SyncIn pulse (CTR = PRD), the value of TBPHS = 200 is loaded into the target TBCTR register so the target time-base is always leading the controller's time-base by 120°.

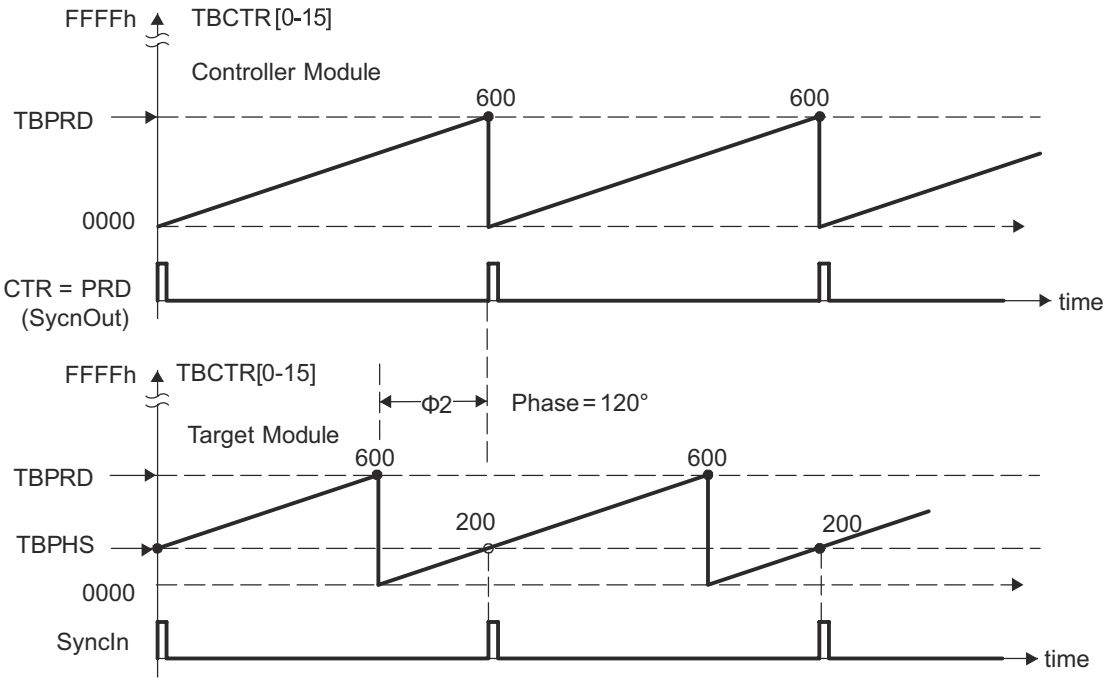


Figure 18-62. Timing Waveforms Associated With Phase Control Between 2 Modules

ADVANCE INFORMATION

18.4 ePWM Module Control and Status Registers

Table 18-22 lists the memory-mapped registers for the ePWM Module Control and Status Registers. All register offset addresses not listed in Table 18-22 should be considered as reserved locations and the register contents should not be modified.

Table 18-22. ePWM Module Control and Status Registers

Offset	Acronym	Register Name	Section
0h	TBSTS	Time-Base Status Register	Section 18.4.1
2h	TBCTL	Time-Base Control Register	Section 18.4.2
4h	TBPHS	Time-Base Phase Register	Section 18.4.3
8h	TBPRD	Time-Base Period Register	Section 18.4.4
Ah	TBCTR	Time-Base Counter Register	Section 18.4.5
Ch	CMPCTL	Counter-Compare Control Register	Section 18.4.6
10h	CMPA	Counter-Compare A Register	Section 18.4.7
14h	AQCTLA	Action-Qualifier Control Register for Output A (EPWMxA)	Section 18.4.8
16h	CMPB	Counter-Compare B Register	Section 18.4.9
18h	AQSFRC	Action-Qualifier Software Force Register	Section 18.4.10
1Ah	AQCTLB	Action-Qualifier Control Register for Output B (EPWMxB)	Section 18.4.11
1Ch	DBCTL	Dead-Band Generator Control Register	Section 18.4.12
1Eh	AQCSFRC	Action-Qualifier Continuous S/W Force Register Set	Section 18.4.13
20h	DBFED	Dead-Band Generator Falling Edge Delay Count Register	Section 18.4.14
22h	DBRED	Dead-Band Generator Rising Edge Delay Count Register	Section 18.4.15
24h	TZDCSEL	Trip Zone Digital Compare Event Select Register	Section 18.4.16
26h	TZSEL	Trip-Zone Select Register	Section 18.4.17
28h	TZEINT	Trip-Zone Enable Interrupt Register	Section 18.4.18
2Ah	TZCTL	Trip-Zone Control Register	Section 18.4.19
2Ch	TZCLR	Trip-Zone Clear Register	Section 18.4.20
2Eh	TZFLG	Trip-Zone Flag Register	Section 18.4.21
30h	ETSEL	Event-Trigger Selection Register	Section 18.4.22
32h	TZFRC	Trip-Zone Force Register	Section 18.4.23
34h	ETFLG	Event-Trigger Flag Register	Section 18.4.24
36h	ETPS	Event-Trigger Pre-Scale Register	Section 18.4.25
38h	ETFRC	Event-Trigger Force Register	Section 18.4.26
3Ah	ETCLR	Event-Trigger Clear Register	Section 18.4.27
3Eh	PCCTL	PWM-Chopper Control Register	Section 18.4.28
60h	DCACTL	Digital Compare A Control Register	Section 18.4.29
62h	DCTRIPSEL	Digital Compare Trip Select Register	Section 18.4.30
64h	DCFCTL	Digital Compare Filter Control Register	Section 18.4.31
66h	DCBCTL	Digital Compare B Control Register	Section 18.4.32
68h	DCFOFFSET	Digital Compare Filter Offset Register	Section 18.4.33
6Ah	DCCAPCTL	Digital Compare Capture Control Register	Section 18.4.34
6Ch	DCFWINDOW	Digital Compare Filter Window Register	Section 18.4.35
6Eh	DCFOFFSETCNT	Digital Compare Filter Offset Counter Register	Section 18.4.36
70h	DCCAP	Digital Compare Counter Capture Register	Section 18.4.37
72h	DCFWINDOWCNT	Digital Compare Filter Window Counter Register	Section 18.4.38

18.4.1 TBSTS Register (Offset = 0h) [reset = 1h]

TBSTS is shown in Figure 18-63 and described in Table 18-23.

Figure 18-63. TBSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					CTRMAX	SYNCI	CTRDIR
R-0h					R-0h	R/W-0h	R-1h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-23. TBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	Reserved
2	CTRMAX	R	0h	Time-Base Counter Max Latched Status Bit 0h = Reading a 0 indicates the time-base counter never reached its maximum value. Writing a 0 will have no effect. 1h = Reading a 1 on this bit indicates that the time-base counter reached the max value 0xFFFF. Writing a 1 to this bit will clear the latched event.
1	SYNCI	R/W	0h	Input Synchronization Latched Status Bit 0h = Writing a 0 will have no effect. Reading a 0 indicates no external synchronization event has occurred. 1h = Reading a 1 on this bit indicates that an external synchronization event has occurred (EPWMxSYNCI). Writing a 1 to this bit will clear the latched event.
0	CTRDIR	R	1h	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]. 0h = Time-Base Counter is currently counting down. 1h = Time-Base Counter is currently counting up.

18.4.2 TBCTL Register (Offset = 2h) [reset = 83h]

TBCTL is shown in Figure 18-64 and described in Table 18-24.

Figure 18-64. TBCTL Register

15	14	13	12	11	10	9	8
FREE_SOFT		PHSDIR	CLKDIV			HSPCLKDIV	
R/W-0h		R/W-0h	R/W-0h			R/W-1h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HSPCLKDIV	SWFSYNC	SYNCOSEL		PRDLD	PHSEN	CTRMODE	
R/W-1h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-3h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-24. TBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the ePWM time-base counter during emulation events: 0h = Stop after the next time-base counter increment or decrement. 1h = Stop when counter completes a whole cycle. In up-count mode, stop when the time-base counter = period (TBCTR = TBPRD). In down-count mode, stop when the time-base counter = 0x0000 (TBCTR = 0x0000). In up-down-count mode, stop when the time-base counter = 0x0000 (TBCTR = 0x0000). 2h = Free run 3h = Free run
13	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. 0h = Count down after the synchronization event. 1h = Count up after the synchronization event.
12-10	CLKDIV	R/W	0h	Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = VCLK4 / (HSPCLKDIV x CLKDIV) 0h = /1 (default on reset) 1h = /2 2h = /4 3h = /8 4h = /16 5h = /32 6h = /64 7h = /128
9-7	HSPCLKDIV	R/W	1h	High Speed Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = VCLK4 / (HSPCLKDIV x CLKDIV) 0h = /1 1h = /2 (default on reset) 2h = /4 3h = /6 4h = /8 5h = /10 6h = /12 7h = /14

Table 18-24. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SWFSYNC	R/W	0h	Software Forced Synchronization Pulse. This event is ORed with the EPWMxSYNCI input of the ePWM module. SWFSYNC is valid (operates) only when EPWMxSYNCI is selected by SYNCOSEL = 0. 0h = Writing a 0 has no effect and reads always return a 0. 1h = Writing a 1 forces a one-time synchronization pulse to be generated.
5-4	SYNCOSEL	R/W	0h	Synchronization Output Select. These bits select the source of the EPWMxSYNCO signal. 0h = 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	PRDLD	R/W	0h	Active Period Register Load From Shadow Register Select 0h = The period register (TBPRD) is loaded from its shadow register when the time-base counter, TBCTR, is equal to 0. A write or read to the TBPRD register accesses the shadow register. 1h = Load the TBPRD register immediately without using a shadow register. A write or read to the TBPRD register directly accesses the active register.
2	PHSEN	R/W	0h	Counter Register Load From Phase Register Enable 0h = Do not load the time-base counter (TBCTR) from the time-base phase register (TBPHS) 1h = 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	CTRMODE	R/W	3h	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: 0h = Up-count mode 1h = Down-count mode 2h = Up-down-count mode 3h = Stop-freeze counter operation (default on reset)

18.4.3 TBPHS Register (Offset = 4h) [reset = 0h]

TBPHS is shown in Figure 18-65 and described in Table 18-25.

Figure 18-65. TBPHS Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-25. TBPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBPHS	R/W	0h	These bits set time-base counter phase of the selected ePWM relative to the time-base that is supplying the synchronization input signal. Valid values: 0-FFFFh 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.

18.4.4 TBPRD Register (Offset = 8h) [reset = 0h]

TBPRD is shown in Figure 18-66 and described in Table 18-26.

Figure 18-66. TBPRD Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRD															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-26. TBPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBPRD	R/W	0h	These bits determine the period of the time-base counter. This sets the PWM frequency. Valid values: 0-FFFFh 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 time-base counter equals 0. 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.

18.4.5 TBCTR Register (Offset = Ah) [reset = 0h]

TBCTR is shown in Figure 18-67 and described in Table 18-27.

Figure 18-67. TBCTR Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBCTR															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-27. TBCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TBCTR	R/W	0h	Reading these bits gives the current time-base counter value. Valid values: 0-FFFFh 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.

18.4.6 CMPCTL Register (Offset = Ch) [reset = 0h]

CMPCTL is shown in Figure 18-68 and described in Table 18-28.

Figure 18-68. CMPCTL Register

15	14	13	12	11	10	9	8
RESERVED						SHDWBFULL	SHDWAFULL
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	SHDWBMODE	RESERVED	SHDWAMODE	LOADBMODE		LOADAMODE	
R-0h	R/W-0h	R-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 18-28. CMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	SHDWBFULL	R	0h	Counter-compare B (CMPB) Shadow Register Full Status Flag. This bit self clears once a load-strobe occurs. 0h = CMPB shadow FIFO not full yet 1h = Indicates the CMPB shadow FIFO is full a CPU write will overwrite current shadow value.
8	SHDWAFULL	R	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. 0h = CMPA shadow FIFO not full yet 1h = Indicates the CMPA shadow FIFO is full, a CPU write will overwrite the current shadow value.
7	RESERVED	R	0h	Reserved
6	SHDWBMODE	R/W	0h	Counter-compare B (CMPB) Register Operating Mode. 0h = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1h = Immediate mode. Only the active compare B register is used. All writes and reads directly access the active register for immediate compare action.
5	RESERVED	R	0h	Reserved
4	SHDWAMODE	R/W	0h	Counter-compare A (CMPA) Register Operating Mode. 0h = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1h = Immediate mode. Only the active compare register is used. All writes and reads directly access the active register for immediate compare action.
3-2	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). 0h = 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 18-28. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	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). 0h = 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)

18.4.7 CMPA Register (Offset = 10h) [reset = 0h]

CMPA is shown in Figure 18-69 and described in Table 18-29.

Figure 18-69. CMPA Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-29. CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CMPA	R/W	0h	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.

18.4.8 AQCTLA Register (Offset = 14h) [reset = 0h]

AQCTLA is shown in Figure 18-70 and described in Table 18-30.

Figure 18-70. AQCTLA Register

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
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 18-30. AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-10	CBD	R/W	0h	Action when the time-base counter equals the active CMPB register and the counter is decrementing. 0h = 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.
9-8	CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0h = 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.
7-6	CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0h = 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.
5-4	CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0h = 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.
3-2	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. 0h = 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 18-30. AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	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. 0h = 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.

18.4.9 CMPB Register (Offset = 16h) [reset = 0h]

CMPB is shown in Figure 18-71 and described in Table 18-31.

Figure 18-71. CMPB Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-31. CMPB Register Field Descriptions

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.

18.4.10 AQSFRC Register (Offset = 18h) [reset = 0h]

AQSFRC is shown in Figure 18-72 and described in Table 18-32.

Figure 18-72. AQSFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RLDCSF		OTSFB	ACTSFB		OTSFA	ACTSFA	
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 18-32. AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-6	RLDCSF	R/W	0h	AQCSFRC Active Register Reload From Shadow Options 0h = 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).
5	OTSFB	R/W	0h	One-Time Software Forced Event on Output B 0h = 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. 1h = Initiates a single s/w forced event
4-3	ACTSFB	R/W	0h	Action when One-Time Software Force B Is invoked. 0h = Does nothing (action disabled) 1h = Clear (low) 2h = Set (high) 3h = Toggle (Low to High, High to Low). Note: This action is not qualified by counter direction (CNT_dir).
2	OTSFA	R/W	0h	One-Time Software Forced Event on Output A 0h = 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). 1h = Initiates a single software forced event
1-0	ACTSFA	R/W	0h	Action When One-Time Software Force A Is Invoked. 0h = Does nothing (action disabled) 1h = Clear (low) 2h = Set (high) 3h = Toggle (Low to High, High to Low). Note: This action is not qualified by counter direction (CNT_dir).

18.4.11 AQCTLB Register (Offset = 1Ah) [reset = 0h]

AQCTLB is shown in Figure 18-73 and described in Table 18-33.

Figure 18-73. AQCTLB Register

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
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 18-33. AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-10	CBD	R/W	0h	Action when the counter equals the active CMPB register and the counter is decrementing. 0h = 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	CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0h = 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.
7-6	CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0h = 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	CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0h = 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	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. 0h = 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 18-33. AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	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. 0h = 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.

18.4.12 DBCTL Register (Offset = 1Ch) [reset = 0h]

DBCTL is shown in Figure 18-74 and described in Table 18-34.

Figure 18-74. DBCTL Register

15	14	13	12	11	10	9	8
HALFCYCLE	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED		IN_MODE		POLSEL		OUT_MODE	
R-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 18-34. DBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	HALFCYCLE	R/W	0h	Half Cycle Clocking Enable Bit: 0h = Full cycle clocking enabled. The dead-band counters are clocked at the TBCLK rate. 1h = Half cycle clocking enabled. The dead-band counters are clocked at TBCLK x 2.
14-6	RESERVED	R	0h	Reserved
5-4	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 . 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. 0h = 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.
3-2	POLSEL	R/W	0h	Polarity Select Control. Bit 3 controls the S3 switch and bit 2 controls the S2 switch shown in . 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. 0h = 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 18-34. DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	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 . This allows you to selectively enable or bypass the dead-band generation for the falling-edge and rising-edge delay. 0h = 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].

18.4.13 AQCSFRC Register (Offset = 1Eh) [reset = 0h]

AQCSFRC is shown in Figure 18-75 and described in Table 18-35.

Figure 18-75. AQCSFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				CSFB		CSFA	
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 18-35. AQCSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-2	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]. 0h = 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	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. 0h = 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

18.4.14 DBFED Register (Offset = 20h) [reset = 0h]

DBFED is shown in Figure 18-76 and described in Table 18-36.

Figure 18-76. DBFED Register

15	14	13	12	11	10	9	8
RESERVED						DEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
DEL							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-36. DBFED Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	DEL	R/W	0h	Falling Edge Delay Count. 10-bit counter.

18.4.15 DBRED Register (Offset = 22h) [reset = 0h]

DBRED is shown in Figure 18-77 and described in Table 18-37.

Figure 18-77. DBRED Register

15	14	13	12	11	10	9	8
RESERVED						DEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
DEL							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-37. DBRED Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	DEL	R/W	0h	Rising Edge Delay Count. 10-bit counter.

18.4.16 TZDCSEL Register (Offset = 24h) [reset = 0h]

TZDCSEL is shown in Figure 18-78 and described in Table 18-38.

Figure 18-78. TZDCSEL Register

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2			DCBEVT1
R-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCBEVT1		DCAEVT2			DCAEVT1		
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 18-38. TZDCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-9	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Selection 0h = 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 = Reserved 7h = Reserved
8-6	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Selection 0h = 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 = Reserved 7h = Reserved
5-3	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Selection 0h = 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 = Reserved 7h = Reserved
2-0	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Selection 0h = 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 = Reserved 7h = Reserved

18.4.17 TZSEL Register (Offset = 26h) [reset = 0h]

TZSEL is shown in Figure 18-79 and described in Table 18-39.

One-Shot (OSHT) Trip-zone enable/disable (bits 15-8). When any of the enabled pins go low, a one-shot trip event occurs for this ePWM module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA and EPWMxB outputs. The one-shot trip condition remains latched until the user clears the condition via the TZCLR register. Cycle-by-Cycle (CBC) Trip-zone enable/disable (bits 7-0). When any of the enabled pins go low, a cycle-by-cycle trip event occurs for this ePWM module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA and EPWMxB outputs. A cycle-by-cycle trip condition is automatically cleared when the time-base counter reaches zero.

Figure 18-79. TZSEL Register

15	14	13	12	11	10	9	8
DCBEVT1	DCAEVT1	OSHT6	OSHT5	OSHT4	OSHT3	OSHT2	OSHT1
R-0h	R-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
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0h	R-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 18-39. TZSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	DCBEVT1	R	0h	Digital Compare Output B Event 1 Select 0h = Disable DCBEVT1 as one-shot-trip source for this ePWM module 1h = Enable DCBEVT1 as one-shot-trip source for this ePWM module
14	DCAEVT1	R	0h	Digital Compare Output A Event 1 Select 0h = Disable DCAEVT1 as one-shot-trip source for this ePWM module 1h = Enable DCAEVT1 as one-shot-trip source for this ePWM module
13	OSHT6	R/W	0h	Trip-zone 6 (/TZ6) Select 0h = Disable /TZ6 as a one-shot trip source for this ePWM module 1h = Enable /TZ6 as a one-shot trip source for this ePWM module
12	OSHT5	R/W	0h	Trip-zone 5 (/TZ5) Select 0h = Disable /TZ5 as a one-shot trip source for this ePWM module 1h = Enable /TZ5 as a one-shot trip source for this ePWM module
11	OSHT4	R/W	0h	Trip-zone 4 (/TZ4) Select 0h = Disable /TZ4 as a one-shot trip source for this ePWM module 1h = Enable /TZ4 as a one-shot trip source for this ePWM module
10	OSHT3	R/W	0h	Trip-zone 3 (/TZ3) Select 0h = Disable /TZ3 as a one-shot trip source for this ePWM module 1h = Enable /TZ3 as a one-shot trip source for this ePWM module
9	OSHT2	R/W	0h	Trip-zone 2 (/TZ2) Select 0h = Disable /TZ2 as a one-shot trip source for this ePWM module 1h = Enable /TZ2 as a one-shot trip source for this ePWM module
8	OSHT1	R/W	0h	Trip-zone 1 (/TZ1) Select 0h = Disable /TZ1 as a one-shot trip source for this ePWM module 1h = Enable /TZ1 as a one-shot trip source for this ePWM module

Table 18-39. TZSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	DCBEVT2	R	0h	Digital Compare Output B Event 2 Select 0h = Disable DCBEVT2 as a CBC trip source for this ePWM module 1h = Enable DCBEVT2 as a CBC trip source for this ePWM module
6	DCAEVT2	R	0h	Digital Compare Output A Event 2 Select 0h = Disable DCAEVT2 as a CBC trip source for this ePWM module 1h = Enable DCAEVT2 as a CBC trip source for this ePWM module
5	CBC6	R/W	0h	Trip-zone 6 (/TZ6) Select 0h = Disable /TZ6 as a CBC trip source for this ePWM module 1h = Enable /TZ6 as a CBC trip source for this ePWM module
4	CBC5	R/W	0h	Trip-zone 5 (/TZ5) Select 0h = Disable /TZ5 as a CBC trip source for this ePWM module 1h = Enable /TZ5 as a CBC trip source for this ePWM module
3	CBC4	R/W	0h	Trip-zone 4 (/TZ4) Select 0h = Disable /TZ4 as a CBC trip source for this ePWM module 1h = Enable /TZ4 as a CBC trip source for this ePWM module
2	CBC3	R/W	0h	Trip-zone 3 (/TZ3) Select 0h = Disable /TZ3 as a CBC trip source for this ePWM module 1h = Enable /TZ3 as a CBC trip source for this ePWM module
1	CBC2	R/W	0h	Trip-zone 2 (/TZ2) Select 0h = Disable /TZ2 as a CBC trip source for this ePWM module 1h = Enable /TZ2 as a CBC trip source for this ePWM module
0	CBC1	R/W	0h	Trip-zone 1 (/TZ1) Select 0h = Disable /TZ1 as a CBC trip source for this ePWM module 1h = Enable /TZ1 as a CBC trip source for this ePWM module

18.4.18 TZEINT Register (Offset = 28h) [reset = 0h]

TZEINT is shown in Figure 18-80 and described in Table 18-40.

Figure 18-80. TZEINT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-40. TZEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R/W	0h	Digital Comparator Output B Event 2 Interrupt Enable 0h = Disabled 1h = Enabled
5	DCBEVT1	R/W	0h	Digital Comparator Output B Event 1 Interrupt Enable 0h = Disabled 1h = Enabled
4	DCAEVT2	R/W	0h	Digital Comparator Output A Event 2 Interrupt Enable 0h = Disabled 1h = Enabled
3	DCAEVT1	R/W	0h	Digital Comparator Output A Event 1 Interrupt Enable 0h = Disabled 1h = Enabled
2	OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 0h = Disable one-shot interrupt generation 1h = Enable Interrupt generation a one-shot trip event will cause a EPWMx_TZINT VIM interrupt.
1	CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 0h = Disable cycle-by-cycle interrupt generation 1h = Enable interrupt generation a cycle-by-cycle trip event will cause an EPWMx_TZINT VIM interrupt.
0	RESERVED	R	0h	Reserved

18.4.19 TZCTL Register (Offset = 2Ah) [reset = 0h]

TZCTL is shown in Figure 18-81 and described in Table 18-41.

Figure 18-81. TZCTL Register

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2		DCBEVT1	
R-0h				R/W-0h		R/W-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DCAEVT2		DCAEVT1		TZB		TZA	
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 18-41. TZCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-10	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Action On EPWMxB: 0h = 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	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Action On EPWMxB: 0h = 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	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Action On EPWMxA: 0h = 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
5-4	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Action On EPWMxA: 0h = 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	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. 0h = 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	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. 0h = 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

18.4.20 TZCLR Register (Offset = 2Ch) [reset = 0h]

TZCLR is shown in Figure 18-82 and described in Table 18-42.

Figure 18-82. TZCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-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 18-42. TZCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R/W1C	0h	Clear Flag for Digital Compare Output B Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCBEVT2 event trip condition.
5	DCBEVT1	R/W1C	0h	Clear Flag for Digital Compare Output B Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCBEVT1 event trip condition.
4	DCAEVT2	R/W1C	0h	Clear Flag for Digital Compare Output A Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCAEVT2 event trip condition.
3	DCAEVT1	R/W1C	0h	Clear Flag for Digital Compare Output A Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCAEVT1 event trip condition.
2	OST	R/W	0h	Clear Flag for One-Shot Trip (OST) Latch 0h = Has no effect. Always reads back a 0. 1h = Clears this Trip (set) condition.
1	CBC	R/W	0h	Clear Flag for Cycle-By-Cycle (CBC) Trip Latch 0h = Has no effect. Always reads back a 0. 1h = Clears this Trip (set) condition.
0	INT	R/W	0h	Global Interrupt Clear Flag. 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. 0h = Has no effect. Always reads back a 0. 1h = Clears the trip-interrupt flag for this ePWM module, TZFLG.INT.

18.4.21 TZFLG Register (Offset = 2Eh) [reset = 0h]

TZFLG is shown in Figure 18-83 and described in Table 18-43.

Figure 18-83. TZFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
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 18-43. TZFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R	0h	Latched Status Flag for Digital Compare Output B Event 2 0h = Indicates no trip event has occurred on DCBEVT2 1h = Indicates a trip event has occurred for the event defined for DCBEVT2
5	DCBEVT1	R	0h	Latched Status Flag for Digital Compare Output B Event 1 0h = Indicates no trip event has occurred on DCBEVT1 1h = Indicates a trip event has occurred for the event defined for DCBEVT1
4	DCAEVT2	R	0h	Latched Status Flag for Digital Compare Output A Event 2 0h = Indicates no trip event has occurred on DCAEVT2 1h = Indicates a trip event has occurred for the event defined for DCAEVT2
3	DCAEVT1	R	0h	Latched Status Flag for Digital Compare Output A Event 1 0h = Indicates no trip event has occurred on DCAEVT1 1h = Indicates a trip event has occurred for the event defined for DCAEVT1
2	OST	R	0h	Latched Status Flag for A One-Shot Trip Event. This bit is cleared by writing the appropriate value to the TZCLR register. 0h = No one-shot trip event has occurred. 1h = Indicates a trip event has occurred on a pin selected as a one-shot trip source.
1	CBC	R	0h	Latched Status Flag for Cycle-By-Cycle Trip Event. This bit is cleared by writing the appropriate value to the TZCLR register. 0h = No cycle-by-cycle trip event has occurred. 1h = 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.

Table 18-43. TZFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	INT	R	0h	Latched Trip Interrupt Status Flag. 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. 0h = Indicates no interrupt has been generated. 1h = Indicates an EPWMx_TZINT VIM interrupt was generated because of a trip condition.

18.4.22 ETSEL Register (Offset = 30h) [reset = 0h]

ETSEL is shown in Figure 18-84 and described in Table 18-44.

Figure 18-84. ETSEL Register

15	14	13	12	11	10	9	8
SOCBEN	SOCBSEL			SOCAEN	SOCASEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED				INTEN	INTSEL		
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 18-44. ETSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SOCBEN	R/W	0h	Enable the ADC Start of Conversion B (EPWMxSOCB) Pulse 0h = Disable EPWMxSOCB 1h = Enable EPWMxSOCB pulse
14-12	SOCBSEL	R/W	0h	EPWMxSOCB Selection Options. These bits determine when a EPWMxSOCB pulse will be generated. 0h = 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.
11	SOCAEN	R/W	0h	Enable the ADC Start of Conversion A (EPWMxSOCA) Pulse 0h = Disable EPWMxSOCA 1h = Enable EPWMxSOCA pulse

Table 18-44. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	SOCASEL	R/W	0h	EPWMxSOCA Selection Options. These bits determine when a EPWMxSOCA pulse will be generated. 0h = 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.
7-4	RESERVED	R	0h	Reserved
3	INTEN	R/W	0h	Enable ePWM Interrupt (EPWMx_INT) Generation 0h = Disable EPWMx_INT generation 1h = Enable EPWMx_INT generation
2-0	INTSEL	R/W	0h	ePWM Interrupt (EPWMx_INT) Selection Options 0h = 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.

18.4.23 TZFRC Register (Offset = 32h) [reset = 0h]

TZFRC is shown in Figure 18-85 and described in Table 18-45.

Figure 18-85. TZFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-45. TZFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R/W	0h	Force Flag for Digital Compare Output B Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCBEVT2 event trip condition and sets the TZFLG[DCBEVT2] bit.
5	DCBEVT1	R/W	0h	Force Flag for Digital Compare Output B Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCBEVT1 event trip condition and sets the TZFLG[DCBEVT1] bit.
4	DCAEVT2	R/W	0h	Force Flag for Digital Compare Output A Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCAEVT2 event trip condition and sets the TZFLG[DCAEVT2] bit.
3	DCAEVT1	R/W	0h	Force Flag for Digital Compare Output A Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCAEVT1 event trip condition and sets the TZFLG[DCAEVT1] bit.
2	OST	R/W	0h	Force a One-Shot Trip Event via Software 0h = Writing of 0 is ignored. Always reads back a 0. 1h = Forces a one-shot trip event and sets the TZFLG[OST] bit.
1	CBC	R/W	0h	Force a Cycle-by-Cycle Trip Event via Software 0h = Writing of 0 is ignored. Always reads back a 0. 1h = Forces a cycle-by-cycle trip event and sets the TZFLG[CBC] bit.
0	RESERVED	R	0h	Reserved

18.4.24 ETFLG Register (Offset = 34h) [reset = 0h]

ETFLG is shown in Figure 18-86 and described in Table 18-46.

Figure 18-86. ETFLG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
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 18-46. ETFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	SOCB	R	0h	Latched ePWM ADC Start-of-Conversion B (EPWMxSOCB) Status Flag 0h = Indicates no EPWMxSOCB event occurred. 1h = 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.
2	SOCA	R	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. 0h = Indicates no event occurred. 1h = 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.
1	RESERVED	R	0h	Reserved
0	INT	R	0h	Latched ePWM Interrupt (EPWMx_INT) Status Flag 0h = Indicates no event occurred. 1h = 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. Refer to Event-Trigger Interrupt Generator figure.

18.4.25 ETPS Register (Offset = 36h) [reset = 0h]

ETPS is shown in Figure 18-87 and described in Table 18-47.

Figure 18-87. ETPS Register

15	14	13	12	11	10	9	8
SOCBCNT		SOCBPRD		SOCACNT		SOCAPRD	
R-0h		R/W-0h		R-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				INTCNT		INTPRD	
R-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 18-47. ETPS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	SOCBCNT	R	0h	ePWM ADC Start-of-Conversion B Event (EPWMxSOCB) Counter Register. These bits indicate how many selected ETSEL[SOCBSEL] events have occurred: 0h = No events have occurred. 1h = 1 event has occurred. 2h = 2 events have occurred. 3h = 3 events have occurred.
13-12	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. 0h = 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
11-10	SOCACNT	R	0h	ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Counter Register. These bits indicate how many selected ETSEL[SOCASEL] events have occurred: 0h = No events have occurred. 1h = 1 event has occurred. 2h = 2 events have occurred. 3h = 3 events have occurred.

Table 18-47. ETPS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	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. 0h = 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	INTCNT	R	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]. 0h = No events have occurred. 1h = 1 event has occurred. 2h = 2 events have occurred. 3h = 3 events have occurred.
1-0	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. 0h = 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)

18.4.26 ETFRC Register (Offset = 38h) [reset = 0h]

ETFRC is shown in Figure 18-88 and described in Table 18-48.

Figure 18-88. ETFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
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 18-48. ETFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	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. 0h = Has no effect. Always reads back a 0. 1h = Generates a pulse on EPWMxSOCB and sets the SOCBFLG bit. This bit is used for test purposes.
2	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. 0h = Writing 0 to this bit will be ignored. Always reads back a 0. 1h = Generates a pulse on EPWMxSOCA and set the SOCAFLG bit. This bit is used for test purposes.
1	RESERVED	R	0h	Reserved
0	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. 0h = Writing 0 to this bit will be ignored. Always reads back a 0. 1h = Generates an interrupt on /EPWMxINT and set the INT flag bit. This bit is used for test purposes.

18.4.27 ETCLR Register (Offset = 3Ah) [reset = 0h]

ETCLR is shown in Figure 18-89 and described in Table 18-49.

Figure 18-89. ETCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
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 18-49. ETCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	SOCB	R/W	0h	ePWM ADC Start-of-Conversion B (EPWMxSOCB) Flag Clear Bit 0h = Writing a 0 has no effect. Always reads back a 0. 1h = Clears the ETFLG[SOCB] flag bit.
2	SOCA	R/W	0h	ePWM ADC Start-of-Conversion A (EPWMxSOCA) Flag Clear Bit 0h = Writing a 0 has no effect. Always reads back a 0. 1h = Clears the ETFLG[SOCA] flag bit.
1	RESERVED	R	0h	Reserved
0	INT	R/W	0h	ePWM Interrupt (EPWMx_INT) Flag Clear Bit 0h = Writing a 0 has no effect. Always reads back a 0. 1h = Clears the ETFLG[INT] flag bit and enable further interrupts pulses to be generated.

18.4.28 PCCTL Register (Offset = 3Eh) [reset = 0h]

PCCTL is shown in Figure 18-90 and described in Table 18-50.

Figure 18-90. PCCTL Register

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

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-50. PCCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	Reserved
10-8	CHPDUTY	R/W	0h	Chopping Clock Duty Cycle 0h = 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 0h = Divide by 1 (no prescale, = 12.5 MHz at 100 MHz VCLK4) 1h = Divide by 2 (6.25 MHz at 100 MHz VCLK4) 2h = Divide by 3 (4.16 MHz at 100 MHz VCLK4) 3h = Divide by 4 (3.12 MHz at 100 MHz VCLK4) 4h = Divide by 5 (2.50 MHz at 100 MHz VCLK4) 5h = Divide by 6 (2.08 MHz at 100 MHz VCLK4) 6h = Divide by 7 (1.78 MHz at 100 MHz VCLK4) 7h = Divide by 8 (1.56 MHz at 100 MHz VCLK4)
4-1	OSHTWTH	R/W	0h	One-Shot Pulse Width 0h = 1 x VCLK4 / 8 wide (= 80 nS at 100 MHz VCLK4) 1h = 2 x VCLK4 / 8 wide (= 160 nS at 100 MHz VCLK4) 2h = 3 x VCLK4 / 8 wide (= 240 nS at 100 MHz VCLK4) 3h = 4 x VCLK4 / 8 wide (= 320 nS at 100 MHz VCLK4) 4h = 5 x VCLK4 / 8 wide (= 400 nS at 100 MHz VCLK4) 5h = 6 x VCLK4 / 8 wide (= 480 nS at 100 MHz VCLK4) 6h = 7 x VCLK4 / 8 wide (= 560 nS at 100 MHz VCLK4) 7h = 8 x VCLK4 / 8 wide (= 640 nS at 100 MHz VCLK4) 8h = 9 x VCLK4 / 8 wide (= 720 nS at 100 MHz VCLK4) 9h = 10 x VCLK4 / 8 wide (= 800 nS at 100 MHz VCLK4) Ah = 11 x VCLK4 / 8 wide (= 880 nS at 100 MHz VCLK4) Bh = 12 x VCLK4 / 8 wide (= 960 nS at 100 MHz VCLK4) Ch = 13 x VCLK4 / 8 wide (= 1040 nS at 100 MHz VCLK4) Dh = 14 x VCLK4 / 8 wide (= 1120 nS at 100 MHz VCLK4) Eh = 15 x VCLK4 / 8 wide (= 1200 nS at 100 MHz VCLK4) Fh = 16 x VCLK4 / 8 wide (= 1280 nS at 100 MHz VCLK4)

Table 18-50. PCCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CHPEN	R/W	0h	PWM-chopping Enable 0h = Disable (bypass) PWM chopping function 1h = Enable chopping function

18.4.29 DCACTL Register (Offset = 60h) [reset = 0h]

DCACTL is shown in Figure 18-91 and described in Table 18-51.

Figure 18-91. DCACTL Register

15	14	13	12	11	10	9	8
RESERVED						EVT2FRC_SYNCSEL	EVT2SRCSEL
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED				EVT1SYNCE	EVT1SOCE	EVT1FRC_SYNCSEL	EVT1SRCSEL
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 18-51. DCACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	EVT2FRC_SYNCSEL	R/W	0h	DCAEVT2 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
8	EVT2SRCSEL	R/W	0h	DCAEVT2 Source Signal Select 0h = Source Is DCAEVT2 Signal 1h = Source Is DCEVTFILT Signal
7-4	RESERVED	R	0h	Reserved
3	EVT1SYNCE	R/W	0h	DCAEVT1 SYNC, Enable/Disable 0h = SYNC Generation Disabled 1h = SYNC Generation Enabled
2	EVT1SOCE	R/W	0h	DCAEVT1 SOC, Enable/Disable 0h = SOC Generation Disabled 1h = SOC Generation Enabled
1	EVT1FRC_SYNCSEL	R/W	0h	DCAEVT1 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
0	EVT1SRCSEL	R/W	0h	DCAEVT1 Source Signal Select 0h = Source Is DCAEVT1 Signal 1h = Source Is DCEVTFILT Signal

18.4.30 DCTRIPSEL Register (Offset = 62h) [reset = 0h]

DCTRIPSEL is shown in Figure 18-92 and described in Table 18-52.

Figure 18-92. DCTRIPSEL Register

15	14	13	12	11	10	9	8
DCBLCOMPSEL				DCBHCOMPSEL			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
DCALCOMPSEL				DCAHCOMPSEL			
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 18-52. DCTRIPSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	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. Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input
11-8	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. Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input
7-4	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. Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input
3-0	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. Values not shown are reserved. If a device does not have a particular comparitor, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input

18.4.31 DCFCTL Register (Offset = 64h) [reset = 0h]

DCFCTL is shown in Figure 18-93 and described in Table 18-53.

Figure 18-93. DCFCTL Register

15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESERVED	RESERVED	PULSESEL		BLANKINV	BLANKE	SRCSEL	
R-0h	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 18-53. DCFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-8	RESERVED	R	0h	Reserved for TI Test
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved for TI Test
5-4	PULSESEL	R/W	0h	Pulse Select For Blanking and Capture Alignment 0h = Time-base counter equal to period (TBCTR = TBPRD) 1h = Time-base counter equal to zero (TBCTR = 0x0000) 2h = Reserved 3h = Reserved
3	BLANKINV	R/W	0h	Blanking Window Inversion 0h = Blanking window not inverted 1h = Blanking window inverted
2	BLANKE	R/W	0h	Blanking Window Enable/Disable 0h = Blanking window is disabled 1h = Blanking window is enabled
1-0	SRCSEL	R/W	0h	Filter Block Signal Source Select 0h = Source Is DCAEVT1 Signal 1h = Source Is DCAEVT2 Signal 2h = Source Is DCBEVT1 Signal 3h = Source Is DCBEVT2 Signal

18.4.32 DCBCTL Register (Offset = 66h) [reset = 0h]

DCBCTL is shown in Figure 18-94 and described in Table 18-54.

Figure 18-94. DCBCTL Register

15	14	13	12	11	10	9	8
RESERVED						EVT2FRC_SYNCSEL	EVT2SRCSEL
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED				EVT1SYNCE	EVT1SOCE	EVT1FRC_SYNCSEL	EVT1SRCSEL
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 18-54. DCBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	EVT2FRC_SYNCSEL	R/W	0h	DCBEVT2 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
8	EVT2SRCSEL	R/W	0h	DCBEVT2 Source Signal Select 0h = Source Is DCBEVT2 Signal 1h = Source Is DCEVTFILT Signal
7-4	RESERVED	R	0h	Reserved
3	EVT1SYNCE	R/W	0h	DCBEVT1 SYNC, Enable/Disable 0h = SYNC Generation Disabled 1h = SYNC Generation Enabled
2	EVT1SOCE	R/W	0h	DCBEVT1 SOC, Enable/Disable 0h = SOC Generation Disabled 1h = SOC Generation Enabled
1	EVT1FRC_SYNCSEL	R/W	0h	DCBEVT1 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
0	EVT1SRCSEL	R/W	0h	DCBEVT1 Source Signal Select 0h = Source Is DCBEVT1 Signal 1h = Source Is DCEVTFILT Signal

18.4.33 DCFOFFSET Register (Offset = 68h) [reset = 0h]

DCFOFFSET is shown in Figure 18-95 and described in Table 18-55.

Figure 18-95. DCFOFFSET Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-55. DCFOFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	OFFSET	R	0h	Blanking Window Offset. Valid values: 0-FFFFh 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.

18.4.34 DCCAPCTL Register (Offset = 6Ah) [reset = 0h]

DCCAPCTL is shown in Figure 18-96 and described in Table 18-56.

Figure 18-96. DCCAPCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						SHDWMODE	CAPE
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 18-56. DCCAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	Reserved
1	SHDWMODE	R/W	0h	TBCTR Counter Capture Shadow Select Mode 0h = 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. 1h = Active Mode. In this mode the shadow register is disabled. CPU reads from the DCCAP register will always return the active register contents.
0	CAPE	R/W	0h	TBCTR Counter Capture Enable/Disable 0h = Disable the time-base counter capture. 1h = Enable the time-base counter capture.

18.4.35 DCFWINDOW Register (Offset = 6Ch) [reset = 0h]

DCFWINDOW is shown in Figure 18-97 and described in Table 18-57.

Figure 18-97. DCFWINDOW Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WINDOW							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-57. DCFWINDOW Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	WINDOW	R/W	0h	Blanking Window Width. Valid values: 0-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. 0h = No blanking window is generated.

18.4.36 DCFOFFSETCNT Register (Offset = 6Eh) [reset = 0h]

DCFOFFSETCNT is shown in Figure 18-98 and described in Table 18-58.

Figure 18-98. DCFOFFSETCNT Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSETCNT															
R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-58. DCFOFFSETCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	OFFSETCNT	R	0h	Blanking Offset Counter. Valid values: 0-FFFFh 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.

18.4.37 DCCAP Register (Offset = 70h) [reset = 0h]

DCCAP is shown in Figure 18-99 and described in Table 18-59.

Figure 18-99. DCCAP Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCCAP															
R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-59. DCCAP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DCCAP	R	0h	Digital Compare Time-Base Counter Capture. Valid values: 0-FFFFh 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.

18.4.38 DCFWINDOWCNT Register (Offset = 72h) [reset = 0h]

DCFWINDOWCNT is shown in Figure 18-100 and described in Table 18-60.

Figure 18-100. DCFWINDOWCNT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WINDOWCNT							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 18-60. DCFWINDOWCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Any writes to these bit(s) must always have a value of 0.
7-0	WINDOWCNT	R	0h	Blanking Window Counter. Valid value: 0-FFh 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.

19 Local Interconnect Network (LIN)

This chapter describes the local interconnect network (LIN) module. Since this module can also operate like a conventional serial communications interface (SCI) port, this module is referred to as the SCI/LIN module in this document. In SCI compatibility mode, this module is functionally compatible to the standalone SCI module. However, since the SCI/LIN module uses a different register/bit structure, code written for this module cannot be directly ported to the standalone SCI module and conversely.

This module can be configured to operate in either SCI (UART) or LIN mode.

19.1 Introduction

The SCI/LIN is compliant to the LIN protocol specified in the *LIN Specification Package*.The SCI/LIN module can be programmed to work either as an SCI or as a LIN. The SCI hardware features are augmented to achieve LIN compatibility.

The SCI module is a universal asynchronous receiver-transmitter (UART) that implements the standard non-return to zero format.

The LIN standard is based on the SCI (UART) serial data link format. The communication concept is single-/multiple- with a message identification for multicast transmission between any network nodes.

Throughout the chapter, compatibility mode refers to SCI mode functionality of the SCI/LIN module. Section 19.2 explains about the SCI functionality and Section 19.3 explains about the LIN functionality. Though the registers are common for LIN and SCI, the register descriptions has notes to identify the register and bit usage in different modes.

19.1.1 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 non-return to zero (NRZ) format
- Double-buffered receive and transmit functions in compatibility mode
- 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 communication mode
- 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
- At 100MHz peripheral clock, 3.125Mbits/s is the maximum baud rate achievable
- Five error flags and seven status flags provide detailed information regarding SCI events
- Two external pins: LINRX and LINTX
- Multibuffered receive and transmit units

Note

The SCI/LIN module is functionally compatible with the C2000™ SCI modules, but not directly software compatible due to different register control structures.

The SCI/LIN module does not support UART hardware flow control. This feature can be implemented in software using a general-purpose I/O pin.

The SCI/LIN module does not support isosynchronous mode as there is no SCICLK pin.

ADVANCE INFORMATION

19.1.2 LIN Features

The following are the features of the LIN module:

- Compatibility with LIN 1.3 protocols
- Configurable baud rate up to 20 kbps
- Two external pins: LINRX and LINTX.
- Multibuffered receive and transmit units
- Identification masks for message filtering
- Automatic header generation
  - Programmable synchronization break field
  - Synchronization field
  - Identifier field
- Automatic Synchronization
  - Synchronization break detection
  - Optional baud rate update
  - Synchronization validation
- Wakeup on LINRX dominant level from transceiver
- Automatic wakeup support
  - Wakeup signal generation
  - Expiration times on wakeup signals
- Automatic bus idle detection
- Error detection
  - Bit error
  - Bus error
  - No-response error
  - Checksum error
  - Synchronization field error
  - Parity error
- 2 interrupt lines with priority encoding for:
  - Receive
  - Transmit
  - ID, error, and status
- Support for LIN 2.0 checksum
- Enhanced synchronizer finite state machine (FSM) support for frame processing
- Enhanced handling of extended frames
- Enhanced baud rate generator
- Update wakeup/go to sleep

19.1.3 LIN Related Collateral

19.1.4 Block Diagram

The SCI/LIN module contains the core SCI block with added sub-blocks to support LIN protocol.

The three major components of the SCI Module are:

- **Transmitter** (TX) contains two major registers to perform the 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 LINTX pin, one bit at a time.
- **Baud Clock Generator**
  - A programmable baud generator produces a baud clock scaled from the input clock VCLK
- **Receiver** (RX) contains two major registers to perform the double-buffering:
  - The receiver shift register (SCIRXSHF) shifts data in from the LINRX 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 their own separate enable and interrupt bits. The receiver and transmitter can each be operated independently or simultaneously in full duplex mode.

To maintain data integrity, the SCI checks the data the SCI 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 19-1 shows the detailed SCI block diagram.

The SCI/LIN module is based on the standalone SCI with the addition of an error detector (parity calculator, checksum calculator, and bit monitor), a mask filter, a synchronizer, and a multibuffered receiver and transmitter. The SCI interface and the baud generator are modified as part of the hardware enhancements for LIN compatibility. Figure 19-2 shows the SCI/LIN block diagram.

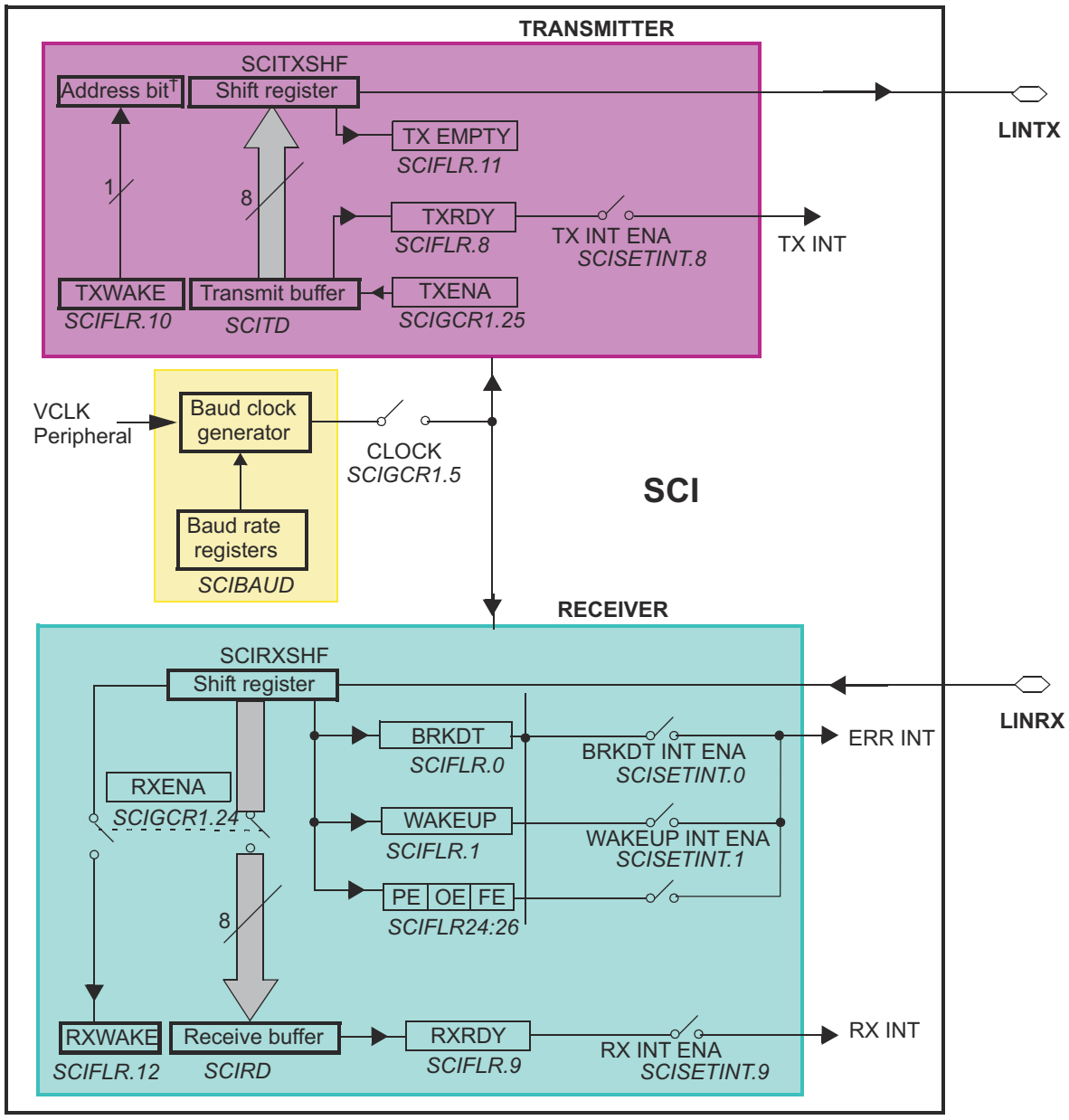


Figure 19-1. SCI Block Diagram

Figure 19-2. SCI/LIN Block Diagram

19.2 Serial Communications Interface Module

19.2.1 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/LIN are user configurable. The configuration options are:

- SCI Frame format
- SCI Timing modes
- SCI Baud rate
- SCI Multiprocessor modes

19.2.1.1 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 19-3.

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 using the PARITY ENA bit. Both examples in Figure 19-3 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 make sure synchronization between communicating devices. Two stop bits are transmitted, if the STOP bit in SCIGCR1 register is set. The examples shown in Figure 19-3 use one stop bit per frame.

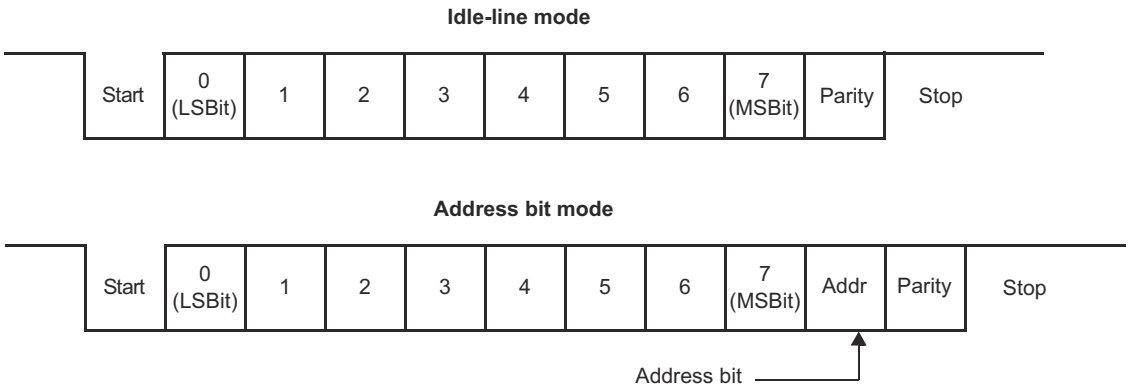


Figure 19-3. Typical SCI Data Frame Formats

19.2.1.2 SCI Asynchronous Timing Mode

The SCI can be configured to use the asynchronous timing mode using TIMING MODE bit in SCIGCR1 register.

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 LINRX pin are of logic level 0. As soon as a falling edge is detected on LINRX, the SCI assumes that a frame is being received and synchronizes itself to the bus.

To prevent interpreting noise as Start bit SCI expects LINRX 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 LINRX 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 19-4 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 LINTX pin. The transmitter then holds the current bit value on LINTX for 16 SCI baud clock periods.

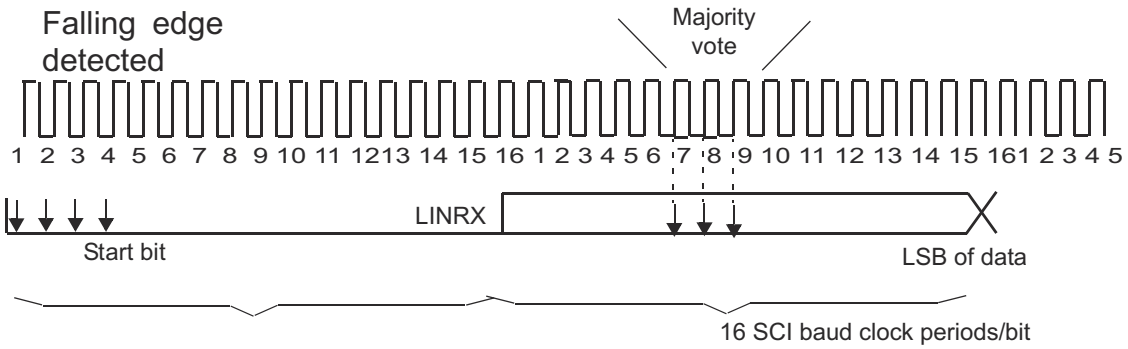


Figure 19-4. Asynchronous Communication Bit Timing

19.2.1.3 SCI Baud Rate

The SCI/LIN has an internally generated serial clock determined by the peripheral VCLK and the prescalers P and M in this register. The SCI uses the 24-bit integer prescaler P value in the BRS register to select the required baud rates. The additional 4-bit fractional divider M refines the baud rate selection.

In asynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$SCICLK\ Frequency = \frac{VCLK\ Frequency}{P + 1 + \frac{M}{16}}$$

$$Asynchronous\ baud\ value = \frac{SCICLK\ Frequency}{16}$$

For P = 0,

$$Asynchronous\ baud\ value = \frac{VCLK\ Frequency}{32}$$

19.2.1.4 SCI Multiprocessor Communication Modes

In some applications, the SCI can be connected to more than one serial communication device. In such a multiprocessor configuration, several frames of data can 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 the devices 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 using 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.

19.2.1.4.1 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 19-5 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:

Step 1: Write a 1 to the TXWAKE bit.

Step 2: 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.

Step 3: Wait for the SCI to clear the TXWAKE flag.

Step 4: Write the address value to SCITD.

As indicated by Step 3, software can wait for the SCI to clear the TXWAKE bit. However, the SCI clears the TXWAKE bit at the same time the SCI 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 the bit.

When idle-line multiprocessor communications are used, software must make sure that the idle time exceeds 10 bit periods before addresses (using one of the methods mentioned above), and software must also make sure 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 results in data interpretation errors by other devices receiving the transmission.

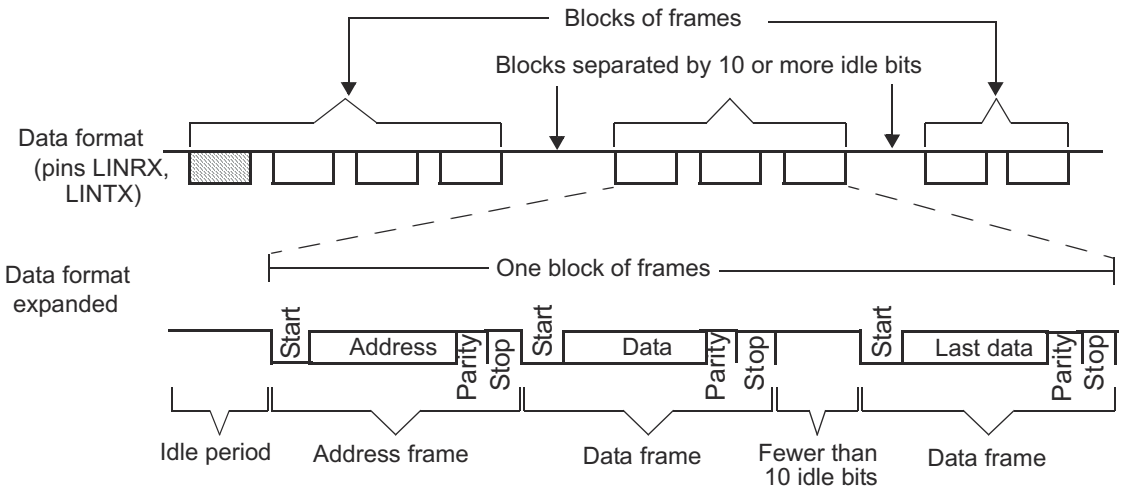


Figure 19-5. Idle-Line Multiprocessor Communication Format

19.2.1.4.2 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 19-6 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.

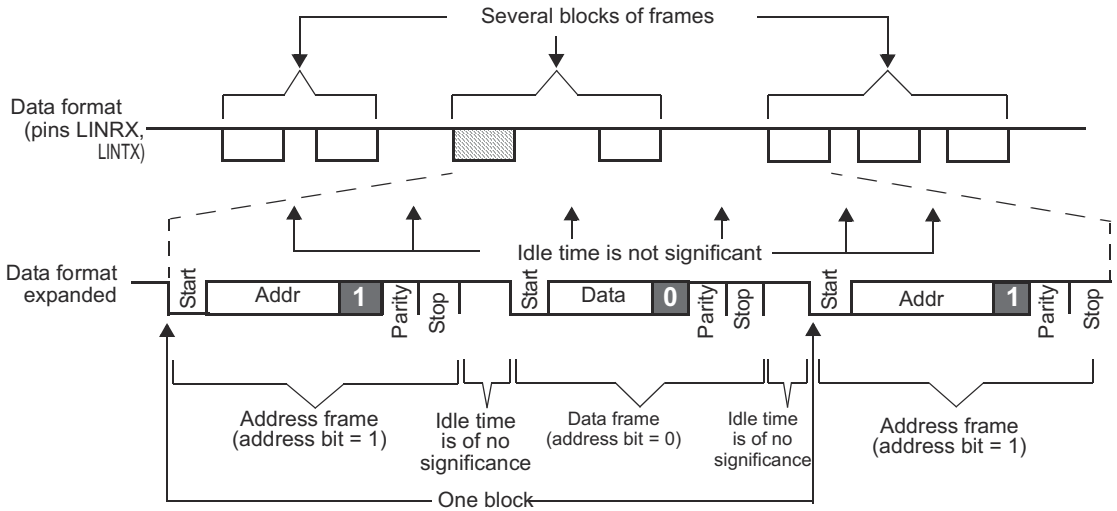


Figure 19-6. Address-Bit Multiprocessor Communication Format

ADVANCE INFORMATION

19.2.1.5 SCI Multibuffered Mode

To reduce CPU load when receiving or transmitting data , the SCI/LIN module has eight separate receive and transmit buffers. Multibuffered mode is enabled by setting the MBUF MODE bit.

The multibuffer 3-bit counter counts the data bytes transferred from the SCIRXSHF register to the RDy receive buffers and TDy transmit buffers register to SCITXSHF register. The 3-bit compare register contains the number of data bytes expected to be received or transmitted. the LENGTH value in SCIFORMAT register indicates the expected length and is used to load the 3-bit compare register.

A receive interrupt (RX interrupt; see the SCIINTVECT0 and SCIINTVECT1 registers), and a receive ready RXRDY flag set in SCIFLR register can occur after receiving a response if there are no response receive errors for the frame (such as, there is, frame error, and overrun error).

A transmit interrupt (TX interrupt), and a transmit ready flag (TXRDY flag in SCIFLR register) can occur after transmitting a response.

Figure 19-7 and Figure 19-8 show the receive and transmit multibuffer functional block diagram, respectively.

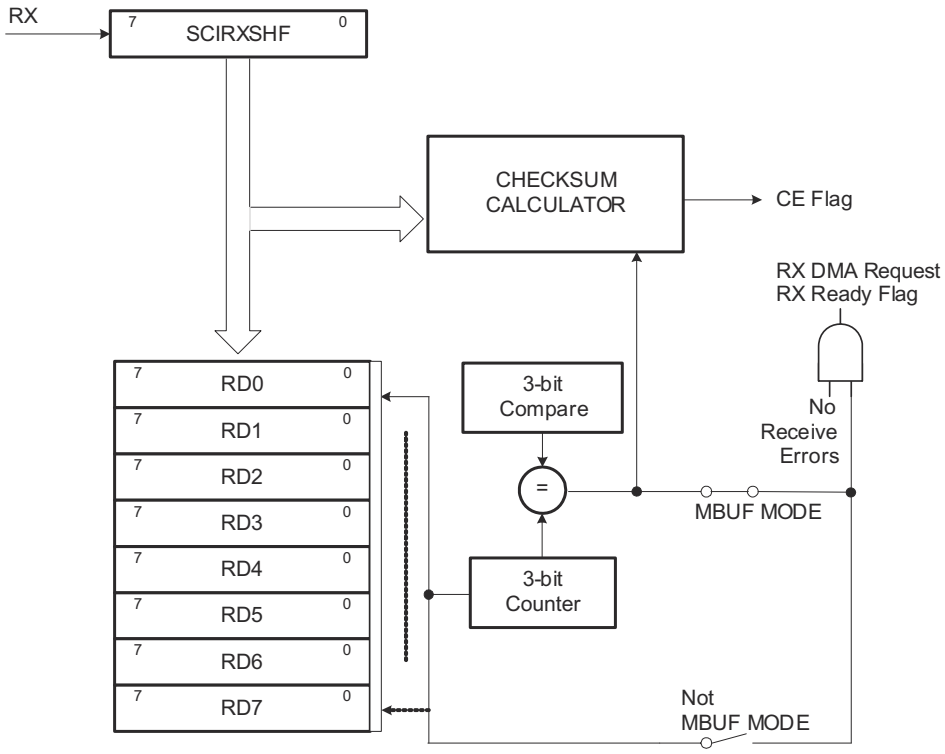


Figure 19-7. Receive Buffers

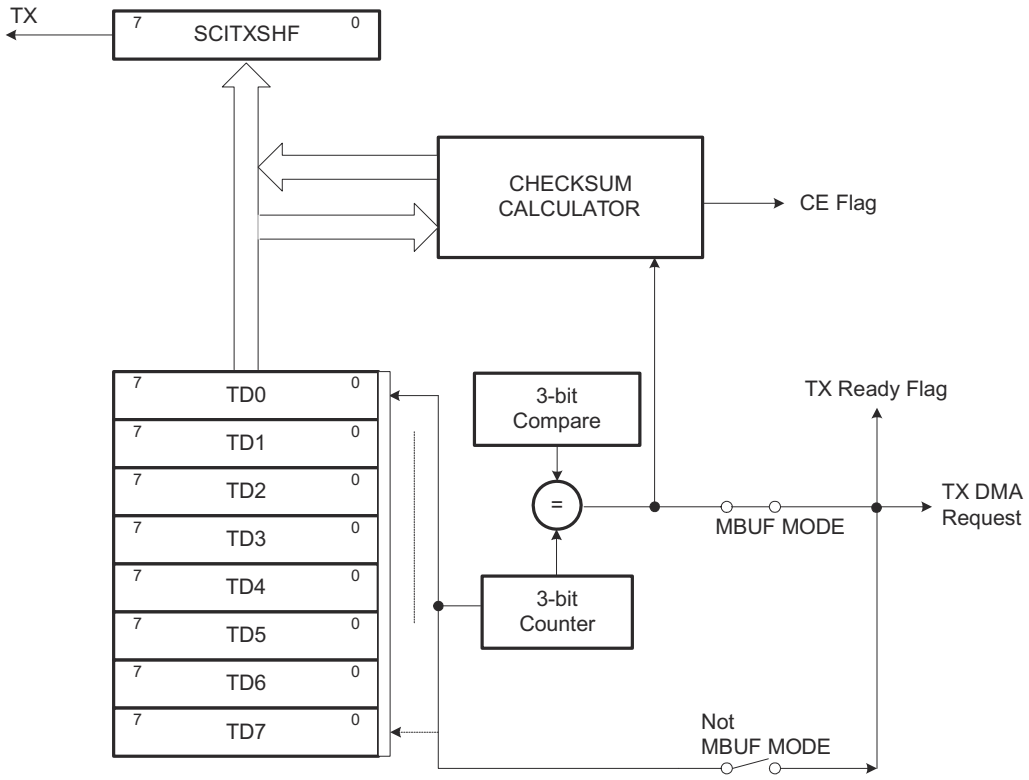


Figure 19-8. Transmit Buffers

ADVANCE INFORMATION

19.2.2 SCI Interrupts

The SCI/LIN module has two interrupt lines, level 0 and level 1, to the vectored interrupt manager (VIM) module (see Figure 19-9). 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.

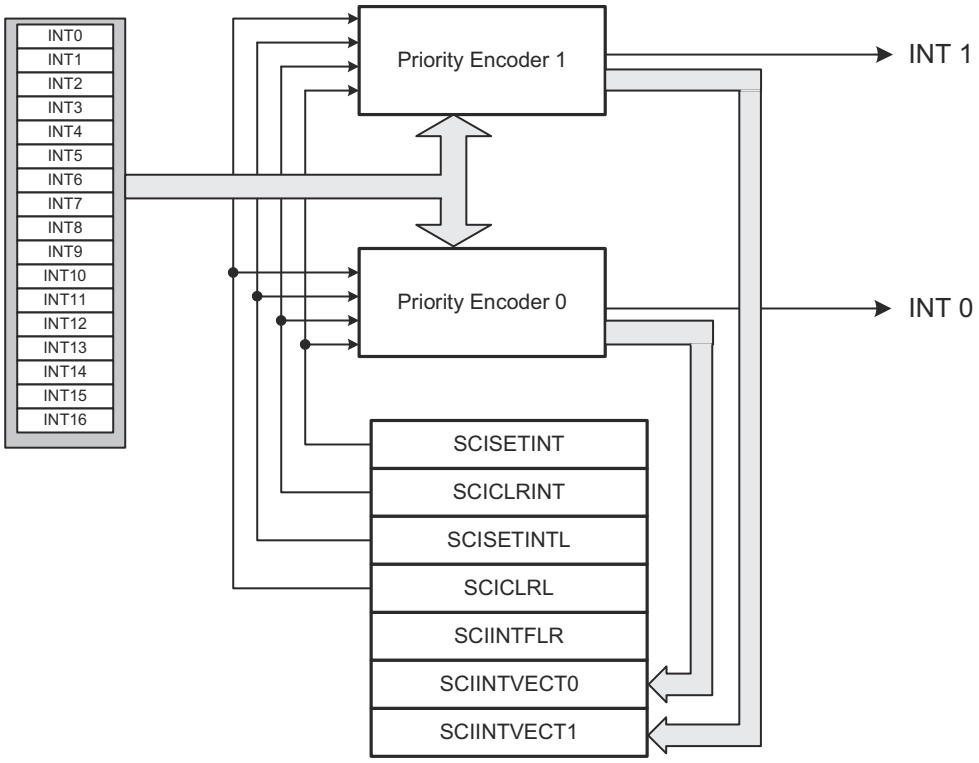


Figure 19-9. General Interrupt Scheme

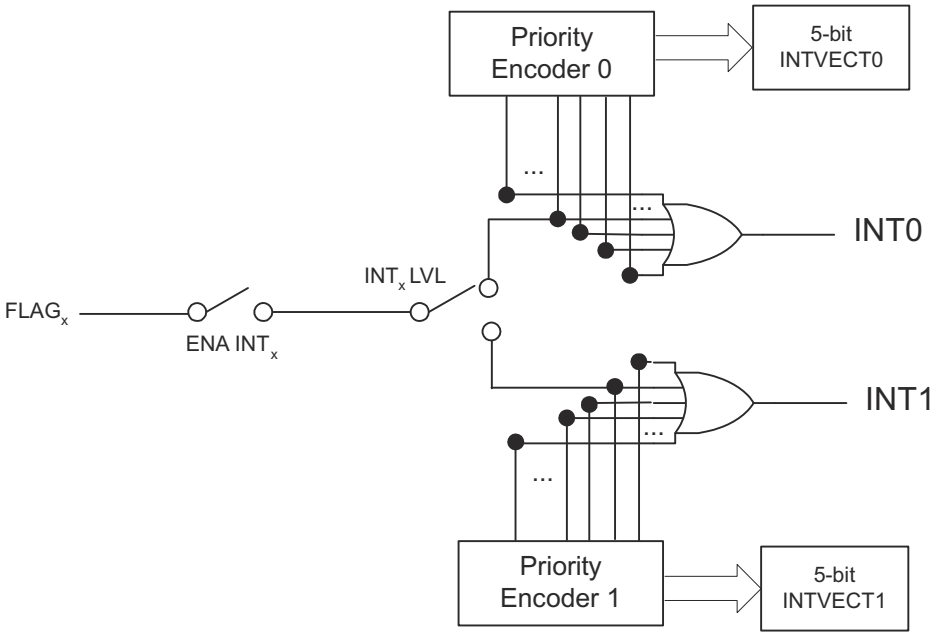


Figure 19-10. Interrupt Generation for Given Flags

19.2.2.1 Transmit Interrupt

To use transmit interrupt functionality, SET TX INT bit must be enabled . 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 . Transmit Interrupt is generated only after the first transfer from SCITD to SCITXSHF, that is first data has to be written to SCITD before any interrupt gets generated. To transmit further data, data can be written 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 using the TXENA bit, by a software reset SWnRST, or by a device hardware reset.

19.2.2.2 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.

19.2.2.3 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 interrupt is triggered once WAKEUP flag is set.

19.2.2.4 Error Interrupts

The following error detections are supported with an interrupt by the SCI module:

- Parity errors (PE)
- Frame errors (FE)
- Break Detect errors (BRKDT)
- Overrun errors (OE)
- Bit errors (BE)

There are 16 interrupt sources in the SCI/LIN module. In SCI mode, 8 interrupts are supported, as listed in Table 19-1.

If all of these errors (PE, FE, BRKDT, OE, BE) are flagged, an interrupt for the flagged errors is 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 (Table 19-2 and Table 19-3).

Table 19-1. SCI/LIN Interrupts

Offset[1]	Interrupt	Applicable to SCI	Applicable to LIN
0	No interrupt	-	-
1	Wakeup	Yes	Yes
2	Inconsistent-sync-field error (ISFE)	No	Yes
3	Parity error (PE)	Yes	Yes
4	ID	No	Yes
5	Physical bus error (PBE)	No	Yes
6	Frame error (FE)	Yes	Yes
7	Break detect (BRKDT)	Yes	No
8	Checksum error (CE)	No	Yes
9	Overrun error (OE)	Yes	Yes
10	Bit error (BE)	Yes	Yes
11	Receive	Yes	Yes
12	Transmit	Yes	Yes
13	No-response error (NRE)	No	Yes
14	Timeout after wakeup signal (150ms)	No	Yes
15	Timeout after three wakeup signals (1.5s)	No	Yes
16	Timeout (Bus Idle, 4s)	No	Yes

(1) Offset 1 is the highest priority. Offset 16 is the lowest priority.

Table 19-2. SCI Receiver Status Flags

SCI Flag	Register	Bit	Value After Reset(1)
CE	SCIFLR	29	0
ISFE	SCIFLR	28	0
NRE	SCIFLR	27	0
FE	SCIFLR	26	0
OE	SCIFLR	25	0
PE	SCIFLR	24	0
RXWAKE	SCIFLR	12	0
RXRDY	SCIFLR	9	0
BUSY	SCIFLR	3	0
IDLE	SCIFLR	2	1
WAKEUP	SCIFLR	1	0
BRKDT	SCIFLR	0	0

(1) The flags are frozen with the reset value while SWnRST = 0.

Table 19-3. SCI Transmitter Status Flags

SCI Flag	Register	Bit	Value After Reset(1)
BE	SCIFLR	31	0
PBE	SCIFLR	30	0
TXWAKE	SCIFLR	10	0
TXEMPTY	SCIFLR	11	1
TXRDY	SCIFLR	8	1

(1) The flags are frozen with the reset value while SWnRST = 0.

19.2.3 SCI Configurations

Before the SCI sends or receives data, the SCI registers can 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 in the SCIGCR0 register is set to 1. Of particular importance is the SWnRST bit in the SCIGCR1 register. The SWnRST is an active-low bit initialized to 0 and keeps the SCI in a reset state until the bit is programmed to 1. Therefore, all SCI configuration can be completed before a 1 is written to the SWnRST bit.

The following list details the configuration steps that software can perform prior to the transmission or reception of data. As long as the SWnRST bit is cleared to 0 the entire time that the SCI is being configured, the order in which the registers are programmed is not important.

- Enable SCI by setting the RESET bit to 1.
- Clear the SWnRST bit to 0 before SCI is configured.
- Select the desired frame format by programming the SCIGCR1 register.
- Select the baud rate to be used for communication by programming the BRS register.
- Set the CLOCK bit in SCIGCR1 to 1 to select the internal clock.
- Set the CONT bit in SCIGCR1 to 1 to make SCI not halt for an emulation breakpoint until the current reception or transmission is complete (this bit is used only in an emulation environment).
- Set the LOOP BACK bit in SCIGCR1 to 1 to connect the transmitter to the receiver internally (this feature is used to perform a self-test).
- Set the RXENA bit in SCIGCR1 to 1, if data is to be received.
- Set the TXENA bit in SCIGCR1 to 1, if data is to be transmitted.
- Set the SWnRST bit to 1 after SCI is configured.
- Perform receiving or transmitting data (see Section 19.2.3.1 or Section 19.2.3.2).

19.2.3.1 Receiving Data

SCI module can receive data in one of the following modes:

- Single-Buffer (Normal) Mode
- Multibuffer Mode

After a valid idle period is detected, data is automatically received as the data arrives on the LINRX pin.

19.2.3.1.1 Receiving Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUF MODE bit in SCIGCR1 is cleared to 0. In this mode, SCI sets the RXRDY bit when the SCI 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 the FE, OE, or PE flags if any of these error conditions were detected in the received data. These error conditions are supported with configurable interrupt capability. The wakeup and break-detect status bits are also set if one of these errors occurs, but the bits do not necessarily occur at the same time that new data is being loaded into SCIRD.

You can receive data by:

1. Polling Receive Ready Flag
2. Receive Interrupt

In polling method, software can poll for the RXRDY bit and read the data from the SCIRD register once the RXRDY bit is set high. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use To use the interrupt method, the SET RX INT bit is set.

19.2.3.1.2 Receiving Data in Multibuffer Mode

Multibuffer mode is selected when the MBUFMODE bit in SCIGCR1 is set to 1. In this mode, SCI sets the RXRDY bit after receiving the programmed number of data in the receive buffer, the complete frame. The error condition detection logic is similar to the single-buffer mode, except that this logic monitors for the complete frame. Like single-buffer mode, use the polling or interrupt method to read the data. The SCI clears the RXRDY bit after the new data in SCIRD has been read.

19.2.3.2 Transmitting Data

The SCI transmitter is enabled if both the TX FUNC bit and the TXENA bit are set to 1. If the TX FUNC bit is not set, the LINTX 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 module can transmit data in one of the following modes:

- Single-Buffer (Normal) Mode
- Multibuffered or Buffered SCI Mode

19.2.3.2.1 Transmitting Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUF MODE bit in SCIGCR1 is cleared to 0. In this mode, SCI waits for data to be written to SCITD, transfers the data to SCITXSHF, and transmits the data. The TXRDY and TX EMPTY bits 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.

You can transmit data by:

1. Polling Transmit Ready Flag
2. Transmit Interrupt

In polling method, software can poll for the TXRDY bit to go high before writing the data to the SCITD register. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use the interrupt method. To use the interrupt method, the SET TX INT bit 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 request is generated, if enabled. Because all data has been transmitted, the interrupt request must be halted. This can either be done by disabling the transmit interrupt (CLR TX INT) 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.

19.2.3.2.2 Transmitting Data in Multibuffer Mode

Multibuffer mode is selected when the MBUF MODE bit in SCIGCR1 is set to 1. Like single-buffer mode, you can use the polling or interrupt method to write the data to be transmitted. The transmitted data has to be written to the SCITD registers. SCI waits for data to be written to the SCITD register and transfers the programmed number of bytes to SCITXSHF to transmit one by one automatically.

19.2.4 SCI Low-Power Mode

The SCI/LIN 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/LIN. During global low-power mode, all clocks to the SCI/LIN 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/LIN 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/LIN 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 wakeup interrupt is used to allow the SCI to exit low-power mode automatically when a low level is detected on the LINRX pin and also this clears the POWERDOWN bit. If wakeup interrupt is disabled, then the SCI/LIN immediately enters low-power mode whenever it is requested and also any activity on the LINRX pin does not cause the SCI to exit low-power mode.

Note

Enabling Local Low-Power Mode During Receive and Transmit

If the wakeup interrupt is enabled and low-power mode is requested while the receiver is receiving data, then the SCI immediately generates a wakeup interrupt to clear the powerdown bit and prevents the SCI from entering low-power mode and thus completes the current reception. Otherwise, if the wakeup interrupt is disabled, then the SCI completes the current reception and then enters the low-power mode.

19.2.4.1 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 can be enhanced to provide selective indication of new data. When SCI receives an address frame that does not match the 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 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 loads 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 can 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 the bit is set to 0 if SCIRD contains data. If the RXWAKE bit is set, then software can check the address in SCIRD against their own address. If SCIRD is still being addressed, then sleep mode can remain disabled; otherwise, the SLEEP bit can 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. Otherwise, these interrupts 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 19-2) 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 can not change the value of the SLEEP bit and can continue to poll RXRDY.

19.3 Local Interconnect Network Module

19.3.1 LIN Communication Formats

The SCI/LIN module can be used in LIN mode or SCI mode. The enhancements for baud generation and additional receive/transmit buffers necessary for LIN mode operation are also part of the enhanced buffered SCI module. LIN mode is selected by enabling LIN MODE bit in SCIGCR1 register.

Note

The SCI/LIN is built around the SCI platform and uses a similar sampling scheme: 16 samples for each bit with majority vote on samples 8, 9, and 10. For the START bit, the first three samples are used.

The SCI/LIN control registers are located at the SCI/LIN base address.

19.3.1.1 LIN Standards

For compatibility with LIN2.0 standard the following additional features are implemented over LIN1.3:

1. Support for LIN 2.0 checksum
2. Enhanced synchronizer FSM support for frame processing
3. Enhanced handling of extended frames
4. Enhanced baud rate generator
5. Update wakeup/go to sleep

The LIN module covers the CPU performance-consuming features, defined in the *LIN Specification Package* Revision 1.3 and 2.0 by hardware.

19.3.1.2 Message Frame

The LIN protocol defines a message frame format, shown in Figure 19-11. Each frame includes one header, one response, one in-frame response space, and inter-byte spaces. In-frame-response and inter-byte spaces can be 0.

There is no arbitration in the definition of the LIN protocol; therefore, multiple nodes responding to a header can be detected as an error.

The LIN bus is a single-channel wired-AND bus. The bus has a binary level: either dominant for a value of 0 or recessive for a value of 1.

Figure 19-11. LIN Protocol Message Frame Format: Header and Response

19.3.1.2.1 Message Header

The header of a message is initiated by a (see Figure 19-12) and consists of a three field-sequence:

- The synchronization break field signaling the beginning of a message
- The synchronization field conveying bit rate information of the LIN bus
- The identification field denoting the content of a message

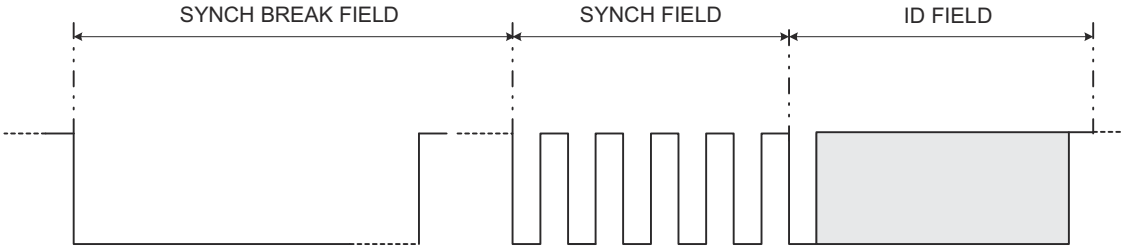


Figure 19-12. Header 3 Fields: Synch Break, Synch, and ID

19.3.1.2.2 Response

The format of the response is as illustrated in Figure 19-13. There are two types of fields in a response: data and checksum. The data field consists of exactly one data byte, one start bit, and one stop bit, for a total of 10 bits. The LSB is transmitted first. The checksum field consists of one checksum byte, one start bit and one stop bit. The checksum byte is the inverted modulo-256 sum over all data bytes in the data fields of the response.

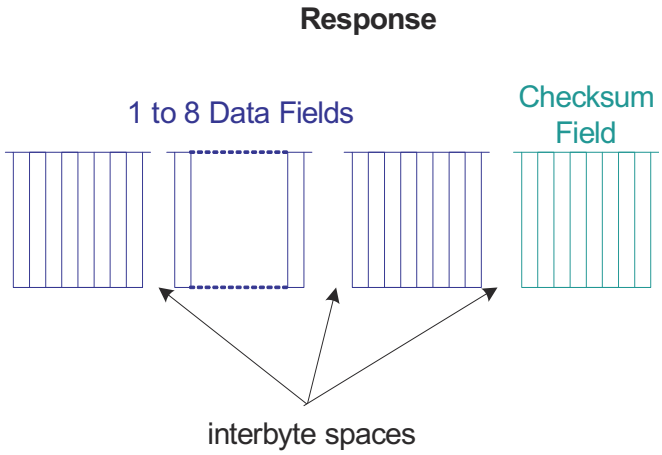


Figure 19-13. Response Format of LIN Message Frame

The format of the response is a stream of N data fields and one checksum field. Typically N is from 1 to 8, with the exception of the extended command frames (Section 19.3.1.6). The length N of the response is indicated either with the optional length control bits of the ID Field (this is used in standards earlier than LIN 1.x); see Table 19-4, or by LENGTH value in SCIFORMAT[18:16] register; see Table 19-5. The SCI/LIN module supports response lengths from 1 to 8 bytes in compliance with LIN 2.0.

Table 19-4. Response Length Info Using IDBYTE Field Bits [5:4] for LIN Standards Earlier than v1.3

ID5	ID4	Number of Data Bytes
0	0	2
0	1	2
1	0	4
1	1	8

Table 19-5. Response Length with SCIFORMAT[18:16] Programming

SCIFORMAT[18:16]	Number of Bytes
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

ADVANCE INFORMATION

19.3.1.3 Synchronizer

The synchronizer has three major functions in the messaging between and nodes. It generates the header data stream, it synchronizes to the LIN bus for responding, and it locally detects timeouts. A bit rate is programmed using the prescalers in the BRSR register to match the indicated LIN_speed value in the LIN description file.

The LIN synchronizer performs the following functions: header signal generation, detection and synchronization to message header with optional baud rate adjustment, response transmission timing and timeout control.

The LIN synchronizer is capable of detecting an incoming break and initializing communication at all times.

19.3.1.4 Baud Rate

The transmission baud rate of any node is configured by the CPU at the beginning; this defines the bit time T_{bit}. The bit time is derived from the fields P and M in the baud rate selection register (BRSR).

The ranges for the prescaler values in the BRSR register are:

P = 0, 1, 2, 3, . . . , 2^{24} - 1

M = 0, 1, 2, . . . , 15

The P values in the BRSR register are user programmable. The P and M dividers can be used for both SCI mode and LIN mode to select a baud rate. If the ADAPT bit is set and the LIN is in adaptive baud rate mode, then all these divider values are automatically obtained during header reception when the synchronization field is measured.

The LIN protocol defines baud rate boundaries as:

$1kHz \le F_{LINCLK} \le 20kHz$

All transmitted bits are shifted in and out at T_{bit} periods.

19.3.1.4.1 Fractional Divider

The M field of the BRSR register modifies the integer prescaler P for fine tuning of the baud rate. The M value adds in increments of 1/16 of the P value.

The bit time, T_{bit} is expressed in terms of the VCLK period T_{VCLK} as follows:

For all P other than 0, and all M,

$$T_{bit} = 16\left(P + 1 + \frac{M}{16}\right)T_{VCLK}$$

For P= 0 : $T_{bit} = 32T_{VCLK}$

Therefore, the LINCLK frequency is given by:

$$F_{LINCLK} = \frac{F_{VCLK}}{16(P + 1 + \frac{M}{16})} \quad \text{For all } P \text{ other than zero}$$

$$F_{LINCLK} = \frac{F_{VCLK}}{32} \quad \text{For } P = 0$$

19.3.1.5 Header Generation

Automatic generation of the LIN protocol header data stream is supported without CPU interaction. The CPU triggers a message header generation and the LIN state machine handles the generation itself. A node initiates header generation on the CPU writes to the IDBYTE in the LINID register. The header is always sent by the to initiate a LIN communication and consists of three fields: synchronization break field, synchronization field, and identification field, as seen in Figure 19-14.

Note

The LIN protocol uses the parity bits in the identifier. The control length bits are optional to the LIN protocol.

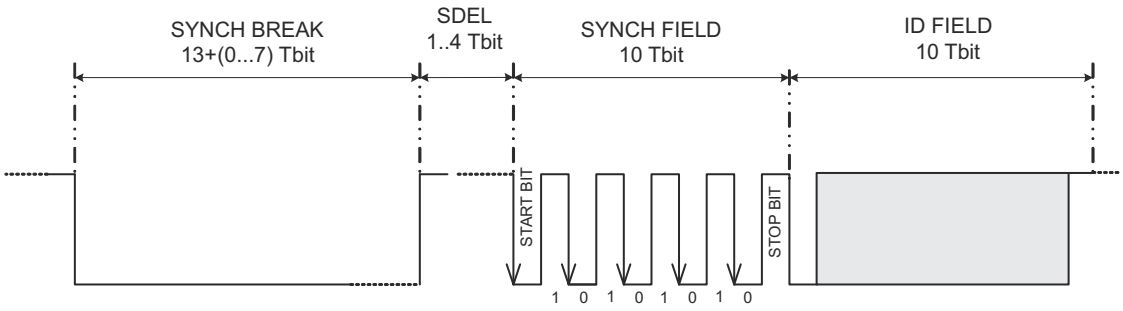


Figure 19-14. Message Header in Terms of T_{bit}

- The break field consists of two components:
  - The synchronization break (SYNCH BREAK) consists of a minimum of 13 (dominant) low bits to a maximum of 20 dominant bits. The sync break length can be extended from the minimum with the 3-bit SBREAK value in the LINCOMP register.
  - The synchronization break delimiter (SDEL) consists of a minimum of 1 (recessive) high bit to a maximum of 4 recessive bits. The delimiter marks the end of the synchronization break field. The sync break delimiter length depends on the 2-bit SDEL value in the LINCOMP register.
- The synchronization field (SYNCH FIELD) consists of one start bit, byte 0x55, and a stop bit. SYNCH FIELD is used to convey T_{bit} information and resynchronize LIN bus nodes.
- The identifier field ID byte can use 6 bits as an identifier, with optional length control and two optional bits as parity of the identifier. The identifier parity is used and checked if the PARITY ENA bit is set. If length control bits are not used, then there can be a total of 64 identifiers plus parity. If neither length control or parity are used there can be up to 256 identifiers. See Figure 19-15 for an illustration of the ID field.

ADVANCE INFORMATION

Note

Optional Control Length Bits

The control length bits only apply to LIN standards prior to LIN 1.3. IDBYTE field conveys response length information if compliant to standards earlier than LIN1.3. The SCIFORMAT register stores the length of the response for later versions of the LIN protocol.

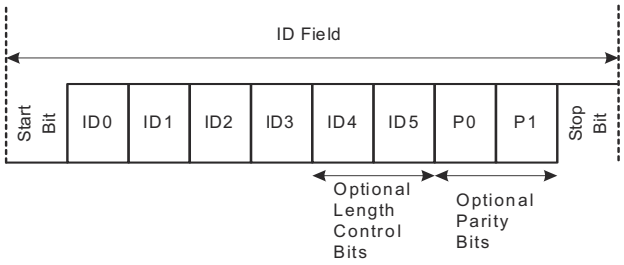


Figure 19-15. ID Field

Note

If the LIN module, configured as a in multibuffer mode, is in the process of transmitting data while a new header comes in, the module can end up responding with the data from the previous interrupted response (not the data corresponding to the new ID). To avoid this scenario, the following procedure can be used:

1. Check for the Bit Error (BE) during the response transmission. If the BE flag is set, this indicates that a collision has happened on the LIN bus (here because of the new Synch Break).
2. In the Bit Error ISR, configure the TD0 and TD1 registers with the next set of data to be transmitted on a TX Match for the incoming ID. Before writing to TD0/TD1 make sure that there was not already an update because of a Bit Error; otherwise, TD0/TD1 can be written twice for one ID.
3. Once the complete ID is received, based on the match, the newly configured data is transmitted by the node.

ADVANCE INFORMATION

19.3.1.5.1 Event Triggered Frame Handling

The LIN 2.0 protocol uses event-triggered frames that can occasionally cause collisions. Event-triggered frames are handled in software.

If no answers to an event triggered frame header, the node sets the NRE flag, and a NRE interrupt occurs if enabled. If a collision occurs, a frame error and checksum error can arise before the NRE error. Those errors are flagged and the appropriate interrupts occur, if enabled.

Frame errors and checksum errors depend on the behavior and synchronization of the responding . If the are totally synchronized and stop transmission once the collision occurred, it is possible that only the NRE error is flagged despite the occurrence of a collision. To detect if there has been a reception of one byte before the NRE error is flagged, the BUS BUSY flag can be used as an indicator.

The BUS BUSY flag is set on the reception of the first bit of the header and remains set until the header reception is complete, and again is set on the reception of the first bit of the response. In the case of a collision, the flag is cleared in the same cycle as the NRE flag is set.

Software can implement the following sequence:

- Once the reception of the header is done (poll for RXID flag), wait for the BUS BUSY flag to get set or the NRE flag to get set.
- If the BUS BUSY flag is not set before the NRE flag, then a true no response is the case (no data has been transmitted onto the bus).
- If the BUS BUSY flag gets set, then wait for the NRE flag to get set or for successful reception. If the NRE flag is set, then a collision has occurred on the bus.

Even in the case of a collision, the received (corrupted) data is accessible in the RX buffers; registers LINRD0 and LINRD1.

19.3.1.5.2 Header Reception and Adaptive Baud Rate

A node baud rate can optionally be adjusted to the detected bit rate as an option to the LIN module. The adaptive baud rate option is enabled by setting the ADAPT bit. During header reception, a measures the baud rate during detection of the synch field. If ADAPT bit is set, then the measured baud rate is compared to the node programmed baud rate and adjusted to the LIN bus baud rate if necessary.

The LIN synchronizer determines two measurements: BRK_count and BAUD_count (Figure 19-16). These values are always calculated during the Header reception for synch field validation (Figure 19-17).

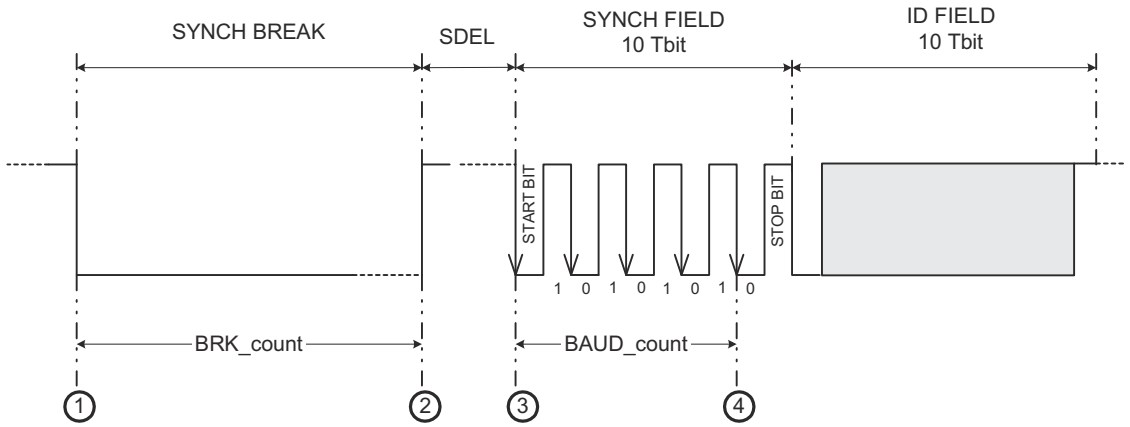


Figure 19-16. Measurements for Synchronization

By measuring the values BRK_count and BAUD_count, a valid sync break sequence can be detected as described in Figure 19-17. The four numbered events in Figure 19-16 signal the start/stop of the synchronizer counter. The synchronizer counter uses VCLK as the time base.

The synchronizer counter is used to measure the sync break relative to the detecting node T_{bit}. For a node receiving the sync break, a threshold of 11 T_{bit} is used as required by the LIN protocol. For detection of the dominant data stream of the sync break, the synchronizer counter is started on a falling edge and stopped on a rising edge of the LINRX. On detection of the sync break delimiter, the synchronizer counter value is saved and then reset.

On detection of five consecutive falling edges, the BAUD_count is measured. Bit timing calculation and consistency to required accuracy is implemented following the recommendations of LIN revision 2.0. A node can calculate a single T_{bit} time by division of BAUD_count by 8. In addition, for consistency between the detected edges the following is evaluated:

BAUD_count + BAUD_count » 2 + BAUD_count » 3 ≤ BRK_count

The BAUD_count value is shifted 3 times to the right and rounded using the first insignificant bit to obtain a T_{bit} unit. If the ADAPT bit is set, then the detected baud rate is compared to the programmed baud rate.

During the header reception processing as illustrated in Figure 19-17, if the measured BRK_count value is less than 11 T_{bit}, the sync break is not valid according to the protocol for a fixed rate. If the ADAPT bit is set, then the MBRS register is used for measuring BRK_count and BAUD_count values and automatically adjusts to any allowed LIN bus rate (refer to *LIN Specification Package 2.0*).

Note

In adaptive mode, the MBRS divider can be set to allow a maximum baud rate that is not more than 10% above the expected operating baud rate in the LIN network. Otherwise, a 0x00 data byte can mistakenly be detected as a sync break.

The break-threshold relative to the node is 11 T_{bit}. The break is 13 T_{bit} as specified in LIN v1.3.

ADVANCE INFORMATION

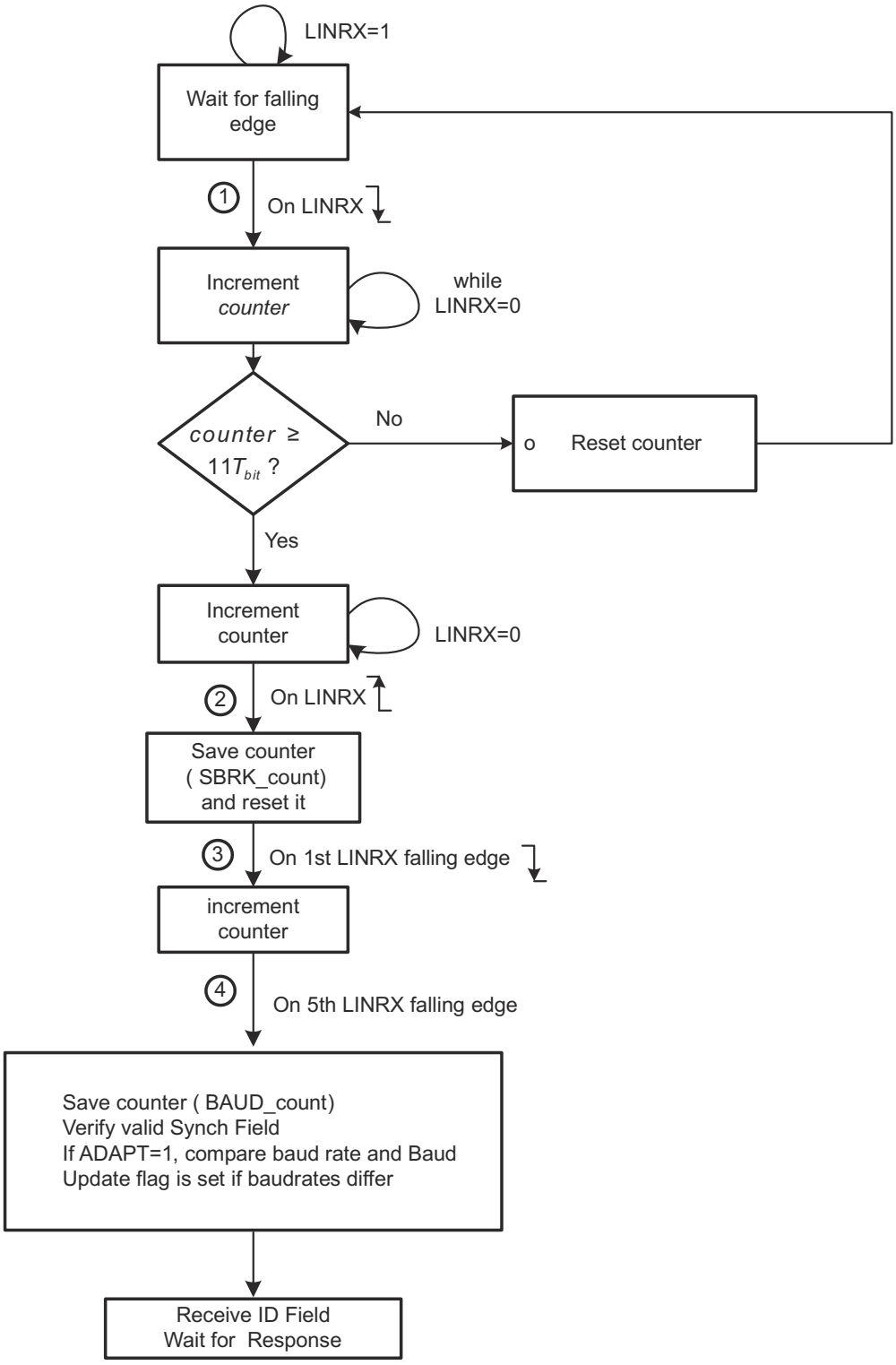


Figure 19-17. Synchronization Validation Process and Baud Rate Adjustment

If the synch field is not detected within the given tolerances, the inconsistent-sync-field-error (ISFE) flag is set. An ISFE interrupt is generated, if enabled by the respective bit in the SCISETINT register. The ID byte can be received after the synch field validation was successful. Any time a valid break (larger than 11 T_{bit}) is detected, the receiver state machine can reset to reception of this new frame. This reset condition is only valid during response state, not if an additional synch break occurs during header reception.

Note

When an inconsistent synch field (ISFE) error occurs, suggested action for the application is to reset the SWnRST bit and set the SWnRST bit to make sure that the internal state machines are back to their normal states.

19.3.1.6 Extended Frames Handling

The LIN protocol 2.0 and prior includes two extended frames with identifiers 62 (user-defined) and 63 (reserved extended). The response data length of the user-defined frame (ID 62, or 0x3E) is unlimited. The length for this identifier is set at network configuration time to be shared with the LIN bus nodes.

Extended frame communication is triggered on reception of a header with identifier 0x3E; see Figure 19-18. Once the extended frame communication is triggered, unlike normal frames, this communication needs to be stopped before issuing another header. To stop the extended frame communication the STOP EXT FRAME bit must be set.

Figure 19-18. Optional Embedded Checksum in Response for Extended Frames

An ID interrupt is generated (if enabled and there is a match) on reception of ID 62 (0x3E). This interrupt allows the CPU using a software counter to keep track of the bytes that are being sent out and decides when to calculate and insert a checksum byte (recommended at periodic rates). To handle this procedure, SC bit is used. A write to the send checksum bit SC initiates an automatic send of the checksum byte. The last data field can always be a checksum in compliance with the LIN protocol.

The periodicity of the checksum insertion, defined at network configuration time, is used by the receiving node to evaluate the checksum of the ongoing message, and has the benefit of enhanced reliability.

For the sending node, the checksum is automatically embedded each time the send checksum bit SC is set. For the receiving node, the checksum is compared each time the compare checksum bit CC is set; see Figure 19-19.

Note

The LIN 2.0 enhanced checksum does not apply to the reserved identifiers. The reserved identifiers always use the classic checksum.

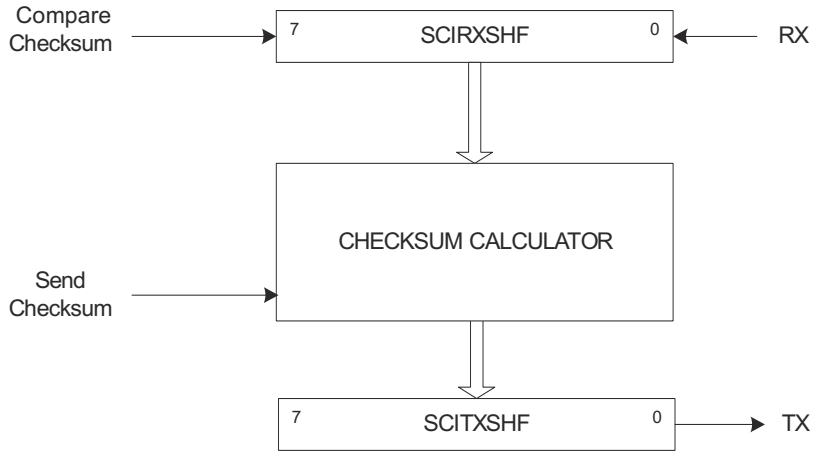


Figure 19-19. Checksum Compare and Send for Extended Frames

19.3.1.7 Timeout Control

Any LIN node listening to the bus and expecting a response initiated from a node can flag a no-response error timeout event. The LIN protocol defines four types of timeout events, which are all handled by the hardware of the LIN module. The four LIN protocol events are:

- No-response timeout error
- Bus idle detection
- Timeout after wakeup signal
- Timeout after three wakeup signals

19.3.1.7.1 No-Response Error (NRE)

The no-response error occurs when any node expecting a response waits for T_{FRAME_MAX} time and the message frame is not fully completed within the maximum length allowed, T_{FRAME_MAX}. After this time, a no-response error (NRE) is flagged in the NRE bit of the SCIFLR register. An interrupt is triggered, if enabled.

As specified in the LIN 1.3 standard, the minimum time to transmit a frame is:

$$T_{FRAME\_MIN} = T_{HEADER\_MIN} + T_{DATA\_FIELD} + T_{CHECKSUM\_FIELD} = 44 + 10N$$

where N = number of data fields.

And the maximum time frame is given by:

$$T_{FRAME\_MAX} = T_{FRAME\_MIN} * 1.4 = (44 + 10N) * 1.4$$

The timeout value T_{FRAME_MAX} is derived from the *N* number of data fields value, see Table 19-6. The *N* value is either embedded in the header ID field for messages or is part of the description file. In the latter case, the 3-bit CHAR value in SCIFORMAT register indicates the value for *N*.

Note

The length coding of the ID field does not apply to two extended frame identifiers, ID fields of 0x3E (62) and 0x3F (63). In these cases, the ID field can be followed by an arbitrary number of data byte fields. Also, the LIN 2.0 protocol specification mentions that ID field 0x3F (63) cannot be used. For these two cases, the NRE is not handled by the LIN hardware.

Table 19-6. Timeout Values in T_{bit} Units

N	T_{DATA_FIELD}	T_{FRAME_MIN}	T_{FRAME_MAX}
1	10	54	76
2	20	64	90
3	30	74	104
4	40	84	118
5	50	94	132
6	60	104	146
7	70	114	160
8	80	124	174

19.3.1.7.2 Bus Idle Detection

The second type of timeout can occur when a node detects an inactive LIN bus: no transitions between recessive and dominant values are detected on the bus. This happens after a minimum of 4 s (this is 80,000 F_{LINCLK} cycles with the fastest bus rate of 20 kbps). If a node detects no activity in the bus as the TIMEOUT bit is set, assume that the LIN bus is in sleep mode. Application software can use the Timeout flag to determine when the LIN bus is inactive and put the LIN into sleep mode by writing the POWERDOWN bit.

Note

After the timeout was flagged, a SWnRESET must be asserted before entering Low-Power Mode. This is required to reset the receiver in case that an incomplete frame is on the bus before the idle period.

19.3.1.7.3 Timeout After Wakeup Signal and Timeout After Three Wakeup Signals

The third and fourth types of timeout are related to the wakeup signal. A node initiating a wakeup must expect a header from the within a defined amount of time: timeout after wakeup signal. See Section 19.4.3 for more details.

19.3.1.8 TXRX Error Detector (TED)

The following sources of error are detected by the TXRX error detector logic (TED). The TED logic consists of a bit monitor, an ID parity checker, and a checksum error. The following errors are detected:

- Bit errors (BE)
- Physical bus errors (PBE)
- Identifier parity errors (PE)
- Checksum errors (CE)

All of these errors (BE, PBE, PE, CE) are flagged. An interrupt for the flagged errors is 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.

ADVANCE INFORMATION

19.3.1.8.1 Bit Errors

A bit error (BE) is detected at the bit time when the bit value that is monitored is different from the bit value that is sent. A bit error is indicated by the BE flag in SCIFLR. After signaling a BE, the transmission is aborted no later than the next byte. The bit monitor makes sure that the transmitted bit in LINTX is the correct value on the LIN bus by reading back on the LINRX pin as shown in Figure 19-20.

Note

If a bit occurs due to receiving a header during a response, NRE/TIMEOUT flag is not set for the new frame.

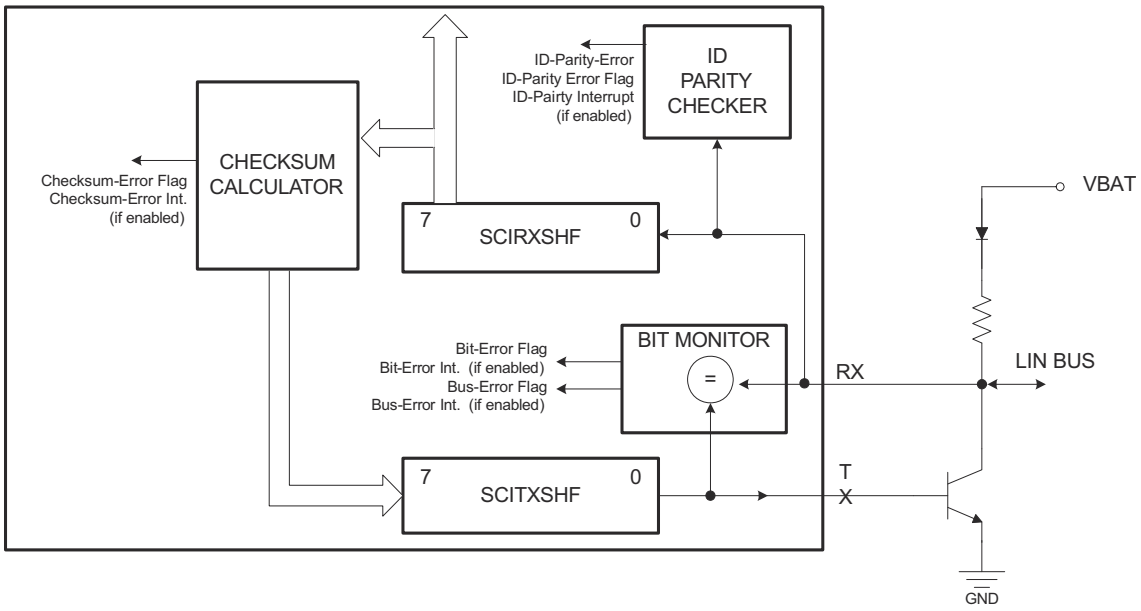


Figure 19-20. TXRX Error Detector

19.3.1.8.2 Physical Bus Errors

A Physical Bus Error (PBE) has to be detected by a , if no valid message can be generated on the bus (bus shorted to GND or VBAT). The bit monitor detects a PBE during the header transmission, if no Synch Break can be generated (for example, because of a bus shortage to VBAT) or if no Synch Break delimiter can be generated (for example, because of a bus shortage to GND). Once the Sync Break Delimiter was validated, all other deviations between the monitored and the sent bit value are flagged as Bit Errors (BE) for this frame.

19.3.1.8.3 ID Parity Errors

If parity is enabled, an ID parity error (PE) is detected if any of the two parity bits of the sent ID byte are not equal to the calculated parity on the receiver node. The two parity bits are generated using the following mixed parity algorithm:

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4 \ (even\ Parity)$$
$$P1 = ID1 \oplus ID3 \oplus ID4 \oplus ID5 \ (odd\ Parity)$$

If an ID-parity error is detected, the ID-parity error is flagged, and the received ID is not valid. See Section 19.3.1.9 for details.

ADVANCE INFORMATION

19.3.1.8.4 Checksum Errors

A checksum error (CE) is detected and flagged at the receiving end, if the calculated modulo-256 sum over all received data bytes (including the ID byte if the enhanced checksum type) plus the checksum byte does not result in 0xFF. The modulo-256 sum is calculated over each byte by adding with carry, where the carry bit of each addition is added to the LSB of the resulting sum.

For the transmitting node, the checksum byte sent at the end of a message is the inverted sum of all the data bytes (see Figure 19-21) for classic checksum implementation. The checksum byte is the inverted sum of the identifier byte and all the data bytes (see Figure 19-22) for the LIN 2.0 compliant enhanced checksum implementation. The classic checksum implementation can always be used for reserved identifiers 60 to 63; therefore, the CTYPE bit is overridden in this case. For signal-carrying-frame identifiers (0 to 59) the type of checksum used depends on the CTYPE bit.

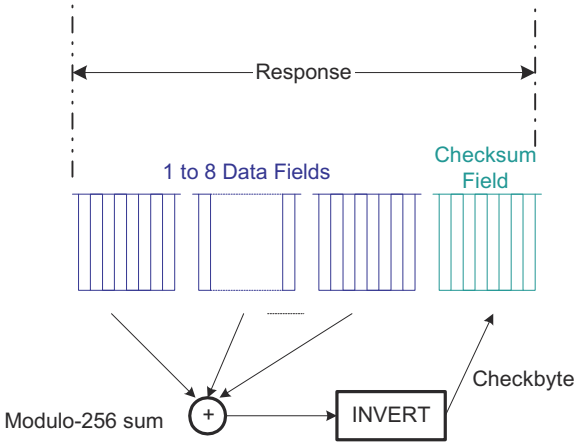


Figure 19-21. Classic Checksum Generation at Transmitting Node

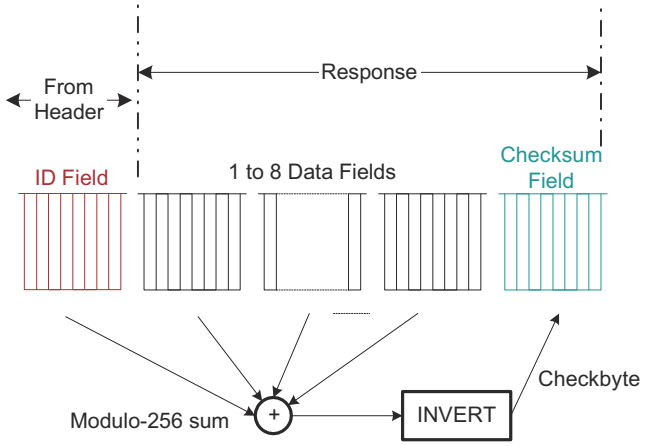


Figure 19-22. LIN 2.0-Compliant Checksum Generation at Transmitting Node

19.3.1.9 Message Filtering and Validation

Message filtering uses the entire identifier to determine which nodes participate in a response, either receiving or transmitting a response. Therefore, two acceptance masks are used as shown in Figure 19-23. During header reception, all nodes filter the ID-Field (ID-Field is the part of the header explained in Figure 19-15) to determine whether the nodes transmit a response or receive a response for the current message. There are two masks for message ID filtering: one to accept a response reception, the other to initiate a response transmission. See Figure 19-23. All nodes compare the received ID to the identifier stored in the ID-TargetTask BYTE of the LINID register and use the RX ID MASK and the TX ID MASK fields in the LINMASK register to filter the bits of the identifier that can not be compared.

If there is an RX match with no parity error and the RXENA bit is set, there is an ID RX flag and an interrupt is triggered if enabled. If there is a TX match with no parity error and the TXENA bit is set, there is an ID TX flag and an interrupt is triggered if enabled in the SCISETINT register.

The masked bits become "don't cares" for the comparison. To build a mask for a set of identifiers, an XOR function can be used.

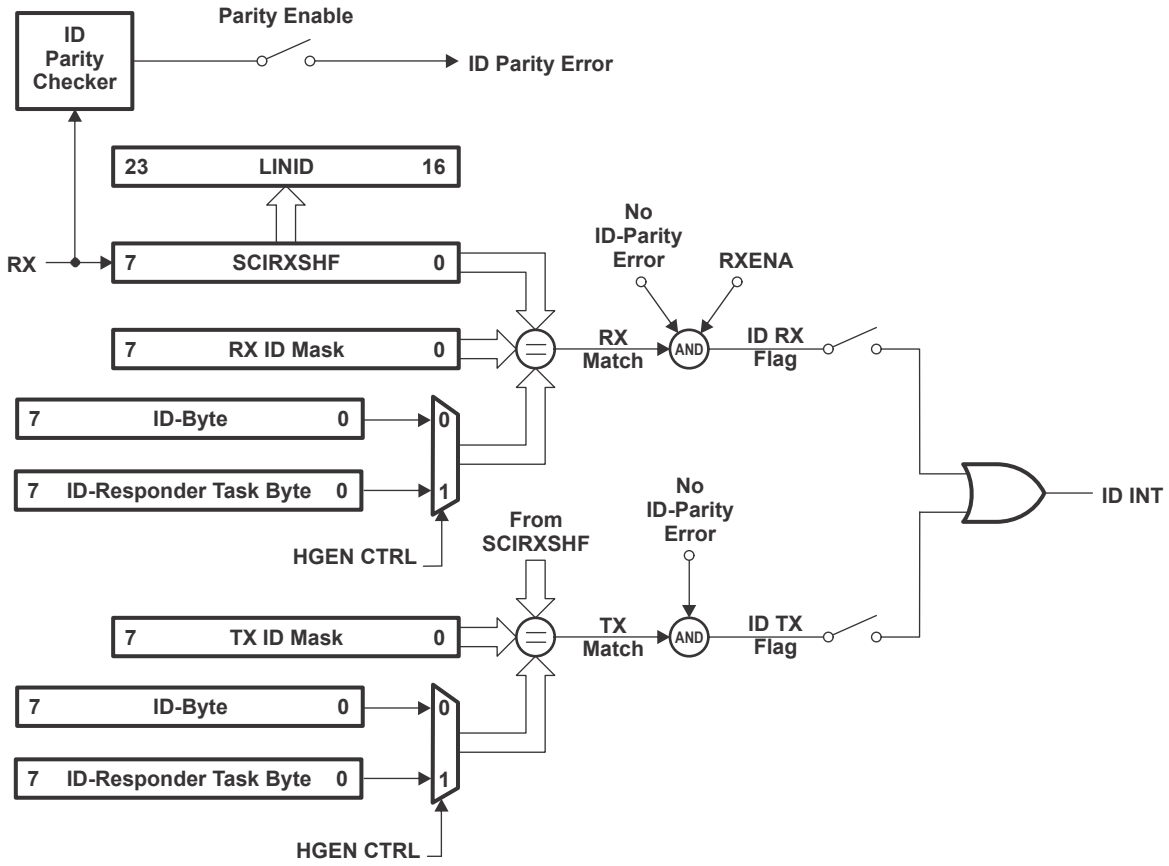


Figure 19-23. ID Reception, Filtering, and Validation

ADVANCE INFORMATION

For example, to build a mask to accept IDs 0x26 and 0x25 using LINID[7:0] = 0x20; that is, compare 5 most-significant bits (MSBs) and filter 3 least-significant bits (LSBs), the acceptance mask can be:

$$(0x26 + 0x25) \oplus 0x20 = 0x07$$

A mask of all zeros compares all bits of the received identifier in the shift register with the ID-BYTE in LINID[7:0]. If HGEN CTRL is set to 1, a mask of 0xFF always causes a match. A mask of all 1s filters all bits of the received identifier, and thus there is an ID match regardless of the content of the ID-TargetTask BYTE field in the LINID register.

Note

When the HGEN CTRL bit = 0, the LIN nodes compare the received ID to the ID-BYTE field in the LINID register, and use the RX ID MASK and the TX ID MASK in the LINMASK register to filter the bits of the identifier that can not be compared.

If there is an RX match with no parity error and the RXENA bit is set, there is an ID RX flag and an interrupt is triggered if enabled. A mask of all 0s compares all bits of the received identifier in the shift register with the ID-BYTE field in LINID[7:0]. A mask of all 1s filters all bits of the received identifier and there is no match.

If HGEN CTLR = 1:

- Received ID is compared with the ID-Target-Task byte, using the RXID mask and the TXID mask.
- A mask of all 1s always result in a match.
- A mask of all 0s means all the bits must be the same to result in a match.
- If a mask has some bits that are 1s, then those bits are not used for the filtering criterion.

If HGEN CTRL = 0:

- Received ID is compared with the ID byte, using the RXID mask and the TXID mask.
- A mask of all 1s results in no match.
- A mask of all 0s means all the bits must be the same to result in a match.
- If a mask has some bits that are 1s, then those bits are not used for the filtering criterion.

During header reception, the received identifier is copied to the Received ID field LINID[23:16]. If there is no parity error and there is either a TX match or an RX match, then the corresponding TX or RX ID flag is set. If the ID interrupt is enabled, then an ID interrupt is generated.

After the ID interrupt is generated, the CPU can read the Received ID field LINID[23:16] and determine what response to load into the transmit buffers.

Note

When byte 0 is written to TD0 (LINTD0[31:24]), the response transmission is automatically generated.

In multibuffer mode, the TXRDY flag is set when all the response data bytes and checksum byte are copied to the shift register SCITXSHF. In non-multibuffer mode, the TXRDY flag is set each time a byte is copied to the SCITXSHF register, and also for the last byte of the frame after the checksum byte is copied to the SCITXSHF register.

In multibuffer mode, the TXEMPTY flag is set when both the transmit buffers TDy and the SCITXSHF shift register are emptied and the checksum has been sent. In non-multibuffer mode, TXEMPTY is set each time TD0 and SCITXSHF are emptied, except for the last byte of the frame where the checksum byte must also be transmitted.

If parity is enabled, all receiving nodes validate the identifier using all eight bits of the received ID byte. The SCI/LIN flags a corrupted identifier if an ID-parity error is detected.

19.3.1.10 Receive Buffers

To reduce CPU load when receiving a LIN N-byte (with N = 1–8) response in interrupt mode , the SCI/LIN module has eight receive buffers. These buffers can store an entire LIN response in the RDy receive buffers. Figure 19-7 illustrates the receive buffers.

The checksum byte following the data bytes is validated by the internal checksum calculator. The checksum error (CE) flag indicates a checksum error and a CE interrupt is generated if enabled in the SCISETINT register.

The multibuffer 3-bit counter counts the data bytes transferred from the SCIRXSHF register to the RDy receive buffers if multibuffer mode is enabled, or to RD0 if multibuffer mode is disabled. The 3-bit compare register contains the number of data bytes expected to be received. In cases where the ID BYTE field does not convey message length (see *Note: Optional Control Length Bits* in Section 19.3.1.5), the LENGTH value, indicates the expected length and is used to load the 3-bit compare register. Whether the length control field or the LENGTH value is used is selectable with the COMM MODE bit.

A receive interrupt, and a receive ready RXRDY flag can occur after receiving a response, if there are no response receive errors for the frame (such as, there is no checksum error, frame error, and overrun error). The checksum byte is compared before acknowledging a reception.

Note

In multibuffer mode following are the scenarios associated with clearing the RXRDY flag bit:

1. The RXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0/1 register.
2. For LENGTH less than or equal to 4, Read to RD0 register clears the RXRDY flag.
3. For LENGTH greater than 4, Read to RD1 register clears the RXRDY flag.

19.3.1.11 Transmit Buffers

To reduce the CPU load when transmitting a LIN N-byte (with N = 1–8) response in interrupt mode , the SCI/LIN module has 8 transmit buffers, TD0–TD7 in LINTD0 and LINTD1. With these transmit buffers, an entire LIN response field can be preloaded in the TXy transmit buffers. Figure 19-8 illustrates the transmit buffers.

The multibuffer 3-bit counter counts the data bytes transferred from the TDy transmit buffers register if multibuffer mode is enabled, or from TD0 to SCITXSHF if multibuffer mode is disabled. The 3-bit compare register contains the number of data bytes expected to be transmitted. If the ID field is not used to convey message length (see *Note: Optional Control Length Bits* in Section 19.3.1.5), the LENGTH value indicates the expected length and is used instead to load the 3-bit compare register. Whether the length control field or the LENGTH value is used is selectable with the COMM MODE bit.

A transmit interrupt (TX interrupt) and a transmit ready flag (TXRDY flag) can occur after transmitting a response.

The checksum byte is automatically generated by the checksum calculator and sent after the data-fields transmission is finished. The multibuffer 3-bit counter counts the data bytes transferred from the TDy buffers into the SCITXSHF register.

Note

The transmit interrupt request can be eliminated until the next series of data is written into the transmit buffers LINTD0 and LINTD1, by disabling the corresponding interrupt using the SCICLRINT register or by disabling the transmitter using the TXENA bit.

19.3.2 LIN Interrupts

LIN and SCI modes have a common interrupt block, as explained in Section 19.2.2. There are 16 interrupt sources in the SCI/LIN module, with 8 of them being LIN mode only, as seen in Table 19-1.

A LIN message frame indicating the timing and sequence of the LIN interrupts that can occur is shown in Figure 19-24.

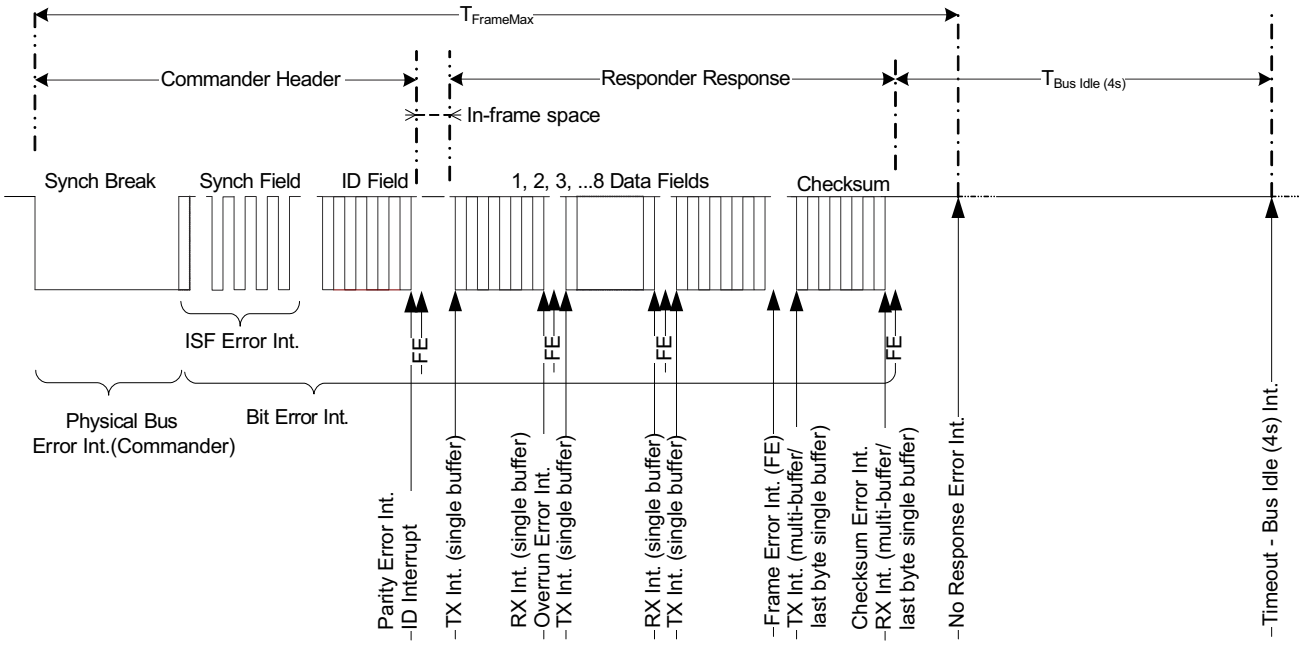


Figure 19-24. LIN Message Frame Showing LIN Interrupt Timing and Sequence

19.3.3 Servicing LIN Interrupts

When servicing an interrupt, clear the corresponding flag in the flag register (SCIFLR). The ISR can follow the guidelines below. This prevents any spurious or duplicate interrupt from occurring.

- Clear the LIN interrupt flag in the SCIFLR register.
- Read the LIN interrupt status register to make sure the flag is cleared.

Note

The transmit interrupt is generated before the LIN transmitter is ready to accept new data. Inside of the LIN transmit ISR, the software can wait until the buffer is completely empty before loading the next data. This can be done by polling for the Bus Busy Flag (SCIFLR.BUSY) to be 0.

ADVANCE INFORMATION

19.3.4 LIN Configurations

The following list details the configuration steps that software can perform prior to the transmission or reception of data in LIN mode. As long as the SWnRST bit in the SCIGCR1 register is cleared to 0 the entire time that the LIN is being configured, the order in which the registers are programmed is not important.

- Enable LIN by setting RESET bit.
- Clear SWnRST to 0 before configuring the LIN.
- Select LIN mode by programming LIN MODE bit.
- Select or mode by programming the CLOCK bit.
- Select the desired frame format (checksum, parity, length control) by programming SCIGCR1.
- Select multibuffer mode by programming MBUF MODE bit.
- Select the baud rate to be used for communication by programming BRSR.
- Set the maximum baud rate to be used for communication by programming MBRSR.
- Set the CONT bit to make LIN not halt for an emulation breakpoint until the LIN 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 if needed (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.
- Select the RX ID MASK and the TX ID MASK fields in the LINMASK register.
- Set SWnRST to 1 after the LIN is configured.
- Perform Receive or Transmit data (see Section 19.3.1.9, Section 19.3.4.1, and Section 19.3.4.2).

19.3.4.1 Receiving Data

The ID RX FLAG is set after a valid LIN ID is received with RX Match. An ID interrupt is generated, if enabled.

19.3.4.1.1 Receiving Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUF MODE bit is cleared to 0. In this mode, LIN sets the RXRDY bit when the LIN transfers newly received data from SCIRXSHF to RD0. The SCI clears the RXRDY bit after the new data in RD0 has been read. Also, as data is transferred from SCIRXSHF to RD0, the LIN sets the FE, OE, or PE flags if any of these error conditions were detected in the received data. These error conditions are supported with configurable interrupt capability.

You can receive data by:

1. Polling Receive Ready Flag
2. Receive Interrupt

In polling method, software can poll for the RXRDY bit and read the data from RD0 byte of the LINRD0 register once the RXRDY bit is set high. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use the interrupt method. To use the interrupt method, the SET RX INT bit is set. If the checksum scheme is enabled by setting the Compare Checksum (CC) bit to 1, the checksum is compared on the byte that is currently being received, which is expected to be the checksum byte. The CC bit is cleared once the checksum is received. A CE is immediately flagged, if there is a checksum error.

19.3.4.1.2 Receiving Data in Multibuffer Mode

Multibuffer mode is selected when the MBUF MODE bit is set to 1. In this mode, LIN sets the RXRDY bit after receiving the programmed number of data in the receive buffer and the checksum field, the complete frame. The error condition detection logic is similar to the single-buffer mode, except that this logic monitors for the complete frame. Like single-buffer mode, you can use the polling or interrupt method to read the data. The received data has to be read from the LINRD0 and LINRD1 registers, based on the number of bytes. For a LENGTH less than or equal to 4, a read from the LINRD0 register clears the RXRDY flag. For a LENGTH greater than 4, a read from the LINRD1 register clears the RXRDY flag. If the checksum scheme is enabled by setting the Compare Checksum (CC) bit to 1 during the reception of the data, then the byte that is received after the reception of the programmed number of data bytes indicated by the LENGTH field is treated as a checksum byte. The CC bit is cleared once the checksum is received and compared.

19.3.4.2 Transmitting Data

The LIN transmitter is enabled if both the TX FUNC bit and the TXENA bit are set to 1. If the TX FUNC bit is not set, the LINTX pin functions as a general-purpose I/O pin rather than as a LIN function pin. Any value written to the TD0 before the TXENA bit is set to 1 is not transmitted. Both of these control bits allow for the LIN transmitter to be held inactive independently of the receiver.

The ID TX flag is set after a valid LIN ID is received with TX Match. An ID interrupt is generated, if enabled.

19.3.4.2.1 Transmitting Data in Single-Buffer Mode

Single-buffer mode is selected when the MBUF MODE bit is cleared to 0. In this mode, LIN waits for data to be written to TD0, transfers the data to SCITXSHF, and transmits the data. The TXRDY and TX EMPTY bits indicate the status of the transmit buffers. That is, when the transmitter is ready for data to be written to TD0, the TXRDY bit is set. Additionally, if both TD0 and SCITXSHF are empty, then the TX EMPTY bit is also set.

You can transmit data by:

1. Polling Transmit Ready Flag
2. Transmit Interrupt

In polling method, software can poll for the TXRDY bit to go high before writing the data to the TD0. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use the interrupt method. To use the interrupt method, the SET TX INT bit is set. When the LIN has completed transmission of all pending frames, the SCITXSHF register and the TD0 are empty, the TXRDY bit is set, and an interrupt request is generated, if enabled. Because all data has been transmitted, the interrupt request can be halted. This can either be done by disabling the transmit interrupt (CLR TX INT) or by disabling the transmitter (clear TXENA bit). If the checksum scheme is enabled by setting the Send Checksum (SC) bit to 1, the checksum byte is sent after the current byte transmission. The SC bit is cleared after the checksum byte has been transmitted.

Note

The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0 or SCIINTVECT1 register.

19.3.4.2.2 Transmitting Data in Multibuffer Mode

Multibuffer mode is selected when the MBUF MODE bit is set to 1. Like single-buffer mode, you can use the polling or interrupt method to write the data to be transmitted. The transmitted data has to be written to the LINTD0 and LINTD1 registers, based on the number of bytes. LIN waits for data to be written to Byte 0 (TD0) of the LINTD0 register and transfers the programmed number of bytes to SCITXSHF to transmit one by one automatically. If the checksum scheme is enabled by setting the Send Checksum (SC) bit to 1, the checksum is sent after transmission of the last byte of the programmed number of data bytes, indicated by the LENGTH field. The SC bit is cleared after the checksum byte has been transmitted.

19.4 Low-Power Mode

The SCI/LIN module 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/LIN module. During global low-power mode, all clocks to the SCI/LIN are turned off so the module is completely inactive. If global low-power mode is requested while the receiver is receiving data, then the SCI/LIN completes the current reception and then enters the low-power mode, that is, module enters low-power mode only when Busy bit (SCIFLR.3) is cleared.

The LIN module can enter low-power mode either when there was no activity on the LINRX pin for more than 4 seconds (this can be either a constant recessive or dominant level) or when a Sleep Command frame was received. Once the Timeout flag (SCIFLR.4) was set or once a Sleep Command was received, the POWERDOWN bit (SCIGCR2.0) must be set by the application software to make the module enter local low-power mode. A wakeup signal terminates the sleep mode of the LIN bus.

Note

Enabling Local Low-Power Mode During Receive and Transmit

If the wakeup interrupt is enabled and low-power mode is requested while the receiver is receiving data, then the SCI/LIN immediately generates a wakeup interrupt to clear the power-down bit. Thus, the SCI/LIN is prevented from entering low-power mode and completes the current reception. Otherwise, if the wakeup interrupt is disabled, the SCI/LIN completes the current reception and then enters the low-power mode.

19.4.1 Entering Sleep Mode

In LIN protocol, a sleep command is used to broadcast the sleep mode to all nodes. The sleep command consists of a diagnostic request frame with identifier 0x3C (60), with the first data field as 0x00. There must be no activity in the bus once all nodes receive the sleep command: the bus is in sleep mode.

Local low-power mode is asserted by setting the POWERDOWN bit; setting this bit stops the clocks to the SCI/LIN internal logic and registers. Clearing the POWERDOWN bit causes SCI/LIN to exit from local low-power mode. All the registers are accessible during local power-down mode. If a register is accessed in low-power mode, this access results in enabling the clock to the module for that particular access alone.

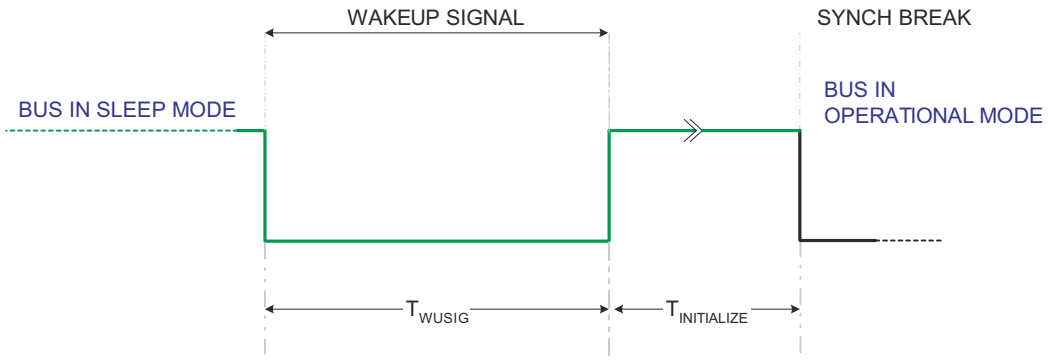
19.4.2 Wakeup

The wakeup interrupt is used to allow the SCI/LIN module to automatically exit a low-power mode. A SCI/LIN wakeup is triggered when a low level is detected on the receive RX pin, and this clears the POWERDOWN bit.

Note

If the wakeup interrupt is disabled, then the SCI/LIN enters low-power mode whenever the SCI/LIN is requested to do so, but a low level on the receive RX pin does not cause the SCI/LIN to exit low-power mode.

In LIN mode, any node can terminate sleep mode by sending a wakeup signal, see Figure 19-25. A node that detects the bus in sleep mode, and with a wakeup request pending, sends a wakeup signal. The wakeup signal is a dominant value on the LIN bus for T_{WUSIG}; this is at least 5 T_{bits} for the LIN bus baud rates. The wakeup signal is generated by sending a 0xF0 byte containing 5 dominant T_{bits} and 5 recessive T_{bits}.



0.25ms ≤ T_{WUSIG} ≤ 5ms

Figure 19-25. Wakeup Signal Generation

Assuming a bus with no noise or loading effects, a write of 0xF0 to TD0 loads the transmitter to meet the wakeup signal timing requirement for T_{WUSIG}. Then, setting the GENWU bit transmits the preloaded value in TD0 for a wakeup signal transmission.

Note

The GENWU bit can be set/reset only when SWnRST is set to 1 and the node is in power-down mode. The bit is cleared on a valid synch break detection. A sending a wakeup request, exits power-down mode upon reception of the wakeup pulse. The bit is cleared on a SWnRST. This can be used to stop a from sending further wakeup requests.

The TI TPIC1021 LIN transceiver, upon receiving a wakeup signal, translates it to the microcontroller for wakeup with a dominant level on the RX pin, or a signal to the voltage regulator. While the POWERDOWN bit is set, if the LIN module detects a recessive-to-dominant edge (falling edge) on the RX pin, the LIN module generates a wakeup interrupt if enabled in the SCISETINT register.

According to LIN protocol 2.0, the TI TPIC1021 LIN transceiver detecting a dominant level on the bus longer than 150ms detects it as a wakeup request. The LIN is ready to listen to the bus in less than 100ms ($T_{INITIALIZE}$<100ms) after a dominant-to-recessive edge (end-of-wakeup signal).

19.4.3 Wakeup Timeouts

The LIN protocol defines the following timeouts for a wakeup sequence. After a wakeup signal has been sent to the bus, all nodes wait for the to send a header. If no synch field is detected before 150ms (3,000 cycles at 20kHz) after a wakeup signal is transmitted, a new wakeup is sent by the same node that requested the first wakeup. This sequence is not repeated more than two times. After three attempts to wake up the LIN bus, wakeup signal generation is suspended for a 1.5s (30,000 cycles at 20kHz) period after three breaks.

Note

To achieve compatibility to LIN1.3 timeout conditions, the MBRS register must be set to make sure that the LIN 2.0 (real-time-based) timings meet the LIN 1.3 bit time base. A node triggering the wakeup can set the MBRS register accordingly to meet the targeted time as 128 Tbits × programmed prescaler.

The LIN handles the wakeup expiration times defined by the LIN protocol with a hardware implementation.

19.5 Emulation Mode

In emulation mode, the CONT bit determines how the SCI/LIN operates when the program is suspended. The SCI/LIN counters are affected by this bit during debug mode. when set, the counters are not stopped and when cleared, the counters are stopped debug mode.

Any reads in emulation mode to a SCI/LIN register do not have any effect on the flags in the SCIFLR register.

Note

When emulation mode is entered during the Frame transmission or reception of the frame and CONT bit is not set, Communication is not expected to be successful. The suggested usage is to set CONT bit during emulation mode for successful communication.

19.6 LIN SCI versus Standard SCI

Table 19-7 compares the LIN/SCI with the standalone-SCI ("standard" SCI) available in all C2000 devices.

Table 19-7. SCI versus LIN-SCI Programming

Standard SCI	LIN SCI
CCS Register Structure	
SciaRegs.REGISTER	APP_LIN.REGISTER
Example Configurations	
Idle Line Mode:	**Idle Line Mode:**
SciaRegs.SCICCR.bit.ADDRIDLE_MODE = 0;	APP_LIN.SCIGCR1.bit.COMMMODE = 0;
Address Bit Mode:	**Address Bit Mode:**
SciaRegs.SCICCR.bit.ADDRIDLE_MODE = 1;	APP_LIN.SCIGCR1.bit.COMMMODE = 1;
FIFO Mode:	**Buffered Mode:**
SciaRegs.SCIFFTX.bit.SCIFFENA = 1;	APP_LIN.SCIGCR1.bit.MBUFMODE = 1;
No FIFO Mode:	**Unbuffered Mode:**
SciaRegs.SCIFFTX.bit.SCIFFENA = 0;	APP_LIN.SCIGCR1.bit.MBUFMODE = 0;

Table 19-7. SCI versus LIN-SCI Programming (continued)

Standard SCI	LIN SCI
Configuration Sequence	
//Into software reset SciaRegs.SCICTL1.bit.SWRESET = 0; *(Configuration instructions)* *//Relinquish SCI from reset* SciaRegs.SCICTL1.bit.SWRESET = 1;	*//Into reset* APP_LIN.SCIGCR0.bit.RESET = 0; *//Out of reset* APP_LIN.SCIGCR0.bit.RESET = 1; *//Into software reset* APP_LIN.SCIGCR1.bit.SWnRST = 0; *(Configuration instructions)* *//Bring out of software reset* APP_LIN.SCIGCR1.bit.SWnRST = 1;
Baud Rate Configuration	
Registers: SciaRegs.SCIHBAUD SciaRegs.SCILBAUD	**Registers**: APP_LIN.BRSR
Fields: BRR = (SciaRegs.SCIHBAUD << 8) + SciaRegs.SCILBAUD	**Fields**: Prescaler(P) = APP_LIN.BRSR.bit.P Fractional Divider(M) = APP_LIN.BRSR.bit.M
Base Clock Rate: $LSPCLK = \frac{SYSCLKOUT}{LowSpeedPrescaler}$	**Base Clock Rate:** $LM\_CLK = \frac{SYSCLKOUT}{2}$
Baud Rate Calculation: $Baud = \frac{LSPCLK}{(BRR+1)\times 8}$	**Baud Rate Calculation:** $Baud = \frac{LM\_CLK}{(P+1+\frac{M}{16})\times 16}$
Basic Transmission	
SciaRegs.SCITXBUF = data;	APP_LIN.SCITD = data;
FIFO/Data Buffer	

Structure:
FIFO
Depth:
8 bits x 16
Fill:
for(iter = 0; iter < depth; iter++)
{
SciaRegs.SCITXBUF = data[i];
}
 	
Empty:
for(iter = 0; iter < depth; iter++)
{
data[i] = SciaRegs.SCIRXBUF.all;
 	
}
 	
	Structure:
	Buffer
	Depth:
	8 bits x 8
	Fill:
	APP_LIN.LINTD0.all = data0123;
	APP_LIN.LINTD1.all = data4567;
	
	Empty:
	data4567 = APP_LIN.LINRD1.all;
	data0123 = APP_LIN.LINRD0.all;
	
	Note: The LINTD0/1 and LINRD0/1 can not be accessed with bytewise operations. Software must pack/unpack data into 32-bit words before writing/reading from the LIN data buffers.

ADVANCE INFORMATION

Table 19-7. SCI versus LIN-SCI Programming (continued)

Standard SCI	LIN SCI
Flags	
Data Reception:	**Data Reception:**
SciaRegs.SCIRXST.bit.RXRDY	APP_LIN.SCIFLR.bit.RXRDY
Transmission Completion:	**Transmission Completion:**
SciaRegs.SCICTL2.bit.TXEMPTY	APP_LIN.SCIFLR.bit.TXEMPTY
Interrupts	
ISR Mapping:	**ISR Mapping:**
PieVectTable.SCIRXINTA = &sciaRxFifoIsr;	PieVectTable.LININT0 = &Lin_Level0_ISR;
PieVectTable.SCITXINTA = &sciaTxFifoIsr;	PieVectTable.LININT1 = &Lin_Level1_ISR;
ISR:	**ISR**:
interrupt void sciaTxFifoIsr(void)	interrupt void Lin_Level0_ISR(void)
{	{
//do TX stuff	*//read-clear interrupt vector*
	LinL0IntVect = APP_LIN.SCIINTVECT0.all;
//clear interrupt flag	
SciaRegs.SCIFFTX.bit.TXFFINTCLR = 1;	if(LinL0IntVect == TXVect)
//acknowledge PIE	{
PieCtrlRegs.PIEACK.all = PIEACK_GROUP9;	*//do TX stuff*
	{
}	if(LinL0IntVect == RXVect)
	{
interrupt void sciaRxFifoIsr(void)	*//do RX stuff*
{	}
//do RX stuff	
	//Acknowledge PIE
// clear overflow flag	PieCtrlRegs.PIEACK.all =
SciaRegs.SCIFFRX.bit.RXFFOVRCLR = 1;	PIEACK_GROUP9;
// clear interrupt flag	}
SciaRegs.SCIFFRX.bit.RXFFINTCLR = 1;	
//acknowledge PIE	interrupt void Lin_Level1_ISR(void)
PieCtrlRegs.PIEACK.all = PIEACK_GROUP9;	{
	//read-clear interrupt vector
}	LinL1IntVect =
	APP_LIN.SCIINTVECT1.all;
	if(LinL1IntVect == TXVect)
	{
	//do TX stuff
	{
	if(LinL1IntVect == RXVect)
	{
	//do RX stuff
	}
	//Acknowledge PIE
	PieCtrlRegs.PIEACK.all = PIEACK_GROUP9;
	}

19.7 SCI/LIN Registers

The SCI/LIN module registers are based on the SCI registers, with added functionality registers enabled by the LIN MODE bit in the SCIGCR1 register.

These registers are accessible in 32-bit reads or writes. The SCI/LIN is controlled and accessed through the registers listed in the following sections. Among the features that can be programmed are the LIN protocol mode, communication and timing modes, baud rate value, frame format, and interrupt configuration.

19.7.1 LIN Base Addresses

Table 19-8. LIN Base Address Table (C28)

Instance	Base Address
APP_LIN	0x0053_0000

19.7.2 APP_LIN Registers

Table 19-9 lists the APP_LIN registers. All register offset addresses not listed in Table 19-9 should be considered as reserved locations and the register contents should not be modified.

Table 19-9. APP_LIN Registers

Offset	Acronym	Register Name	Section
0h	SCIGCR0	Global Control Register 0	Go
4h	SCIGCR1	Global Control Register 1	Go
8h	SCIGCR2	Global Control Register 2	Go
Ch	SCISETINT	Interrupt Enable Register	Go
10h	SCICLEARINT	Interrupt Disable Register	Go
14h	SCISETINTLVL	Set Interrupt Level Register	Go
18h	SCICLEARINTLVL	Clear Interrupt Level Register	Go
1Ch	SCIFLR	Flag Register	Go
20h	SCIINTVECT0	Interrupt Vector Offset Register 0	Go
24h	SCIINTVECT1	Interrupt Vector Offset Register 1	Go
28h	SCIFORMAT	Length Control Register	Go
2Ch	BRSR	Baud Rate Selection Register	Go
30h	SCIED	Emulation buffer Register	Go
34h	SCIRD	Receiver data buffer Register	Go
38h	SCITD	Transmit data buffer Register	Go
3Ch	SCIPIO0	Pin control Register 0	Go
40h	SCIPIO1	Pin control Register 1	Go
44h	SCIPIO2	Pin control Register 2	Go
48h	SCIPIO3	Pin control Register 3	Go
4Ch	SCIPIO4	Pin control Register 4	Go
50h	SCIPIO5	Pin control Register 5	Go
54h	SCIPIO6	Pin control Register 6	Go
58h	SCIPIO7	Pin control Register 7	Go
5Ch	SCIPIO8	Pin control Register 8	Go
60h	LINCOMP	Compare register	Go
64h	LINRD0	Receive data register 0	Go
68h	LINRD1	Receive data register 1	Go
6Ch	LINMASK	Acceptance mask register	Go
70h	LINID	LIN ID Register	Go
74h	LINTD0	Transmit Data Register 0	Go

Table 19-9. APP_LIN Registers (continued)

Offset	Acronym	Register Name	Section
78h	LINTD1	Transmit Data Register 1	Go
7Ch	MBRSR	Maximum Baud Rate Selection Register	Go
90h	IODFTCTRL	IODFT for LIN	Go

Complex bit access types are encoded to fit into small table cells. Table 19-10 shows the codes that are used for access types in this section.

Table 19-10. APP_LIN Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
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.

19.7.2.1 SCIGCR0 Register (Offset = 0h) [reset = 0h]

The SCIGCR0 register defines the module reset.

Figure 19-26. SCIGCR0 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							RESET
R-0h							R/W-0h

Table 19-11. SCIGCR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	RESET	R/W	0h	This bit resets the SCI/LIN module. This bit is effective in LIN or SCI-compatible mode.. This bit affects the reset state of the SCI/LIN module. Reset type: SYSRSn 0h (R/W) = SCI/LIN module is in held in reset. 1h (R/W) = SCI/LIN module is out of reset.

19.7.2.2 SCIGCR1 Register (Offset = 4h) [reset = 0h]

The SCIGCR1 register defines the frame format, protocol, and communication mode used by the SCI.

Figure 19-27. SCIGCR1 Register

31	30	29	28	27	26	25	24
RESERVED						TXENA	RXENA
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED						CONT	LOOPBACK
R-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		STOPEXT FRAME	HGENCTRL	CTYPE	MBUFMODE	ADAPT	SLEEP
R-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
SWnRST	LINMODE	CLK_MASTER	STOP	PARITY	PARITYENA	TIMINGMODE	COMMMODE
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 19-12. SCIGCR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	TXENA	R/W	0h	Transmit enable. This bit is effective in LIN and SCI modes. Data is transferred from SCITD or the TDy (with y=0, 1,...7) buffers in LIN mode to the SCITXSHF shift out register only when the TXENA bit is set. Note: Data written to SCITD or the transmit multibuffer before TXENA is set is not transmitted. If TXENA is cleared while transmission is ongoing, the data previously written to SCITD is sent (including the checksum byte in LIN mode). Reset type: SYSRSn 0h (R/W) = Disable transfers from SCITD or TDy to SCITXSHF 1h (R/W) = Enable transfers of data from SCITD or TDy to SCITXSHF
24	RXENA	R/W	0h	Receive enable. This bit is effective in LIN or SCI-compatible mode. RXENA allows or prevents the transfer of data from SCIRXSHF to SCIRD or the receive multibuffers. Note: Clearing RXENA stops received characters from being transferred into the receive buffer or multibuffers, prevents the RX status flags (see Table 7) 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 the time the reception of a frame is complete, the data from the frame is not transferred into the receive buffer. Note: If RXENA is set before the time the reception of a frame is complete, 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 guaranteed 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 Reset type: SYSRSn 0h (R/W) = Prevents the receiver from transferring data from the shift buffer to the receive buffer or multibuffers 1h (R/W) = Allows the receiver to transfer data from the shift buffer to the receive buffer or multibuffers

ADVANCE INFORMATION

Table 19-12. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-18	RESERVED	R	0h	Reserved
17	CONT	R/W	0h	Continue on suspend. This bit has an effect only when a program is being debugged. The bit determines how the SCI/LIN operates when the program is suspended. This bit affects the LIN counters. When this bit is set, the counters are not stopped during debug. When this bit is cleared, the counters are stopped during debug. Reset type: SYSRSn 0h (R/W) = When debug mode is entered, the SCI/LIN state machine is frozen. Transmissions and LIN counters are halted and resume when debug mode is exited. 1h (R/W) = When debug mode is entered, the SCI/LIN continues to operate until the current transmit and receive functions are complete.
16	LOOPBACK	R/W	0h	Loopback bit. This bit is effective in LIN or SCI-compatible mode. The self-checking option for the SCI/LIN can be selected with this bit. If the LINTX and LINRX pins are configured with SCI/LIN functionality, then the LINTX pin is internally connected to the LINRX pin. Externally, during loop back operation, the LINTX pin outputs a high value and the LINRX pin is in a high-impedance state. If this bit value is changed while the SCI/LIN is transmitting or receiving data, errors may result. Reset type: SYSRSn 0h (R/W) = Loopback mode is disabled. 1h (R/W) = Loopback mode is enabled.
15-14	RESERVED	R	0h	Reserved
13	STOPEXTFRAME	R/W	0h	Stop extended frame communication. This bit is effective in LIN mode only. This bit can be written only during extended frame communication. When the extended frame communication is stopped, this bit is cleared automatically. Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Extended frame communication will be stopped, once current frame transmission/reception is completed.
12	HGENCTRL	R/W	0h	HGEN control bit. This bit is effective in LIN mode only. This bit controls the type of mask filtering comparison. Reset type: SYSRSn 0h (R/W) = ID filtering using ID-Byte. RECEIVEDID and IDBYTE fields in the LINID register are used for detecting a match (using TX/RXMASK values). Mask of 0xFF in LINMASK register will result in NO match. 1h (R/W) = ID filtering using ID-SlaveTask byte (Recommended). RECEIVEDID and IDSLAVETASKBYTE fields in the LINID register are used for detecting a match (using TX/RXMASK values). Mask of 0xFF in LINMASK register will result in ALWAYS match

Table 19-12. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CTYPE	R/W	0h	Checksum type. This bit is effective in LIN mode only. This bit controls the type of checksum to be used: classic or enhanced. Reset type: SYSRSn 0h (R/W) = Classic checksum is used. This checksum is compatible with LIN 1.3 slave nodes. The classic checksum contains the modulo-256 sum with carry over all data bytes. Frames sent with Identifier 60 (0x3C) to 63 (0x3F) must always use the classic checksum. 1h (R/W) = Enhanced checksum is used. The enhanced checksum is compatible with LIN 2.0 and newer slave nodes. The enhanced checksum contains the modulo-256 sum with carry over all data bytes AND the protected Identifier.
10	MBUFMODE	R/W	0h	Multibuffer mode. This bit is effective in LIN or SCI-compatible mode. This bit controls receive/transmit buffer usage, that is, whether the RX/TX multibuffers are used or a single register, RD0/TD0, is used. Reset type: SYSRSn 0h (R/W) = The multibuffer mode is disabled. 1h (R/W) = The multibuffer mode is enabled.
9	ADAPT	R/W	0h	Adapt mode enable. This mode is effective in LIN mode only. This bit has an effect during the detection of the Sync Field. There are two LIN protocol bit rate modes that could be enabled with this bit according to the Node capability file definition: automatic or select. Software and network configuration will decide which of the previous two modes. When this bit is cleared, the LIN 2.0 protocol fixed bit rate should be used. If the ADAPT bit is set, a LIN slave node detecting the baud rate will compare it to the prescalers in BRSR register and update it if they are different. The BRSR register will be updated with the new value. If this bit is not set there will be no adjustment to the BRSR register. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Automatic baud rate adjustment is disabled. 1h (R/W) = Automatic baud rate adjustment is enabled.
8	SLEEP	R/W	0h	SCI sleep. SCI compatibility mode only. In a multiprocessor configuration, this bit controls the receive sleep function. Clearing this bit brings the SCI out of sleep mode. 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. The SLEEP bit is not automatically cleared when an address byte is detected. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Sleep mode is disabled. 1h (R/W) = Sleep mode is enabled.

Table 19-12. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	SWnRST	R/W	0h	Software reset (active low). This bit is effective in LIN or SCI-compatible mode. The SCI/LIN should only be configured while SWnRST = 0. Only the following configuration bits can be changed in runtime (while SWnRESET = 1): - STOP EXT Frame (SCIGCR1[13]) - CC bit (SCIGCR2[17]) - SC bit (SCIGCR2[16]) Reset type: SYSRSn 0h (R/W) = The SCI/LIN is in its reset state no data will be transmitted or received. Writing a 0 to this bit initializes the SCI/LIN state machines and operating flags. All affected logic is held in the reset state until a 1 is written to this bit. 1h (R/W) = The SCI/LIN is in its ready state transmission and reception can occur. After this bit is set to 1, the configuration of the module should not change.
6	LINMODE	R/W	0h	LIN mode This bit controls the mode of operation of the module. Reset type: SYSRSn 0h (R/W) = LIN mode is disabled SCI compatibility mode is enabled. 1h (R/W) = LIN mode is enabled SCI compatibility mode is disabled.
5	CLK_MASTER	R/W	0h	SCI internal clock enable or LIN Master/Slave configuration. In the SCI mode, this bit enables the clock to the SCI module. In LIN mode, this bit determines whether a LIN node is a slave or master. Reset type: SYSRSn 0h (R/W) = SCI-compatible mode: Reserved. LIN mode: The module is in slave mode. 1h (R/W) = SCI-compatible mode: Enable clock to the SCI module. LIN mode: The node is in master mode.
4	STOP	R/W	0h	SCI number of stop bits. This bit is effective in SCI-compatible mode only. 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. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = One stop bit is used. 1h (R/W) = Two stop bits are used.
3	PARITY	R/W	0h	SCI parity odd/even selection. This bit is effective in SCI-compatible mode only. If the PARITY ENA bit (SCIGCR1.2) is set, PARITY designates odd or even parity. 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. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Odd parity is used. 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. 1h (R/W) = Even parity is used. 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.

Table 19-12. SCIGCR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	PARITYENA	R/W	0h	Parity enable. Enables or disables the parity function. Reset type: SYSRSn 0h (R/W) = SCI-compatible mode: Parity disabled no parity bit is generated during transmission or is expected during reception. LIN mode: ID-parity verification is disabled. 1h (R/W) = SCI compatible mode: Parity enabled. A parity bit is generated during transmission and is expected during reception. LIN mode: ID-parity verification is enabled.
1	TIMINGMODE	R/W	0h	SCI timing mode bit. This bit is effective in SCI-compatible mode only. It must be set to 1 when the SCI mode is used. This bit configures the SCI for asynchronous operation. Reset type: SYSRSn 0h (R/W) = Reserved. 1h (R/W) = Must be set to 1 when module is configured for SCI operation
0	COMMMODE	R/W	0h	SCI/LIN communication mode bit. In compatibility mode, it selects the SCI communication mode. In LIN mode it selects length control option for ID-field bits ID4 and ID5. Reset type: SYSRSn 0h (R/W) = SCI-compatible mode: Idle-line mode is used. LIN mode: ID4 and ID5 are not used for length control. 1h (R/W) = SCI-compatible mode: Address-bit mode is used. LIN mode: ID4 and ID5 are used for length control.

19.7.2.3 SCIGCR2 Register (Offset = 8h) [reset = 0h]

The SCIGCR2 register is used to send or compare a checksum byte during extended frames, to generate a wakeup and for low-power mode control of the LIN module.

Figure 19-28. SCIGCR2 Register

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

Table 19-13. SCIGCR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	CC	R/W	0h	Compare Checksum. This mode is effective in LIN mode only. This bit is used by the receiver for extended frames to trigger a checksum compare. The user will initiate this transaction by writing a one to this bit. In non-multibuffer mode, once the CC bit is set, the checksum will be compared on the byte that is currently being received, expected to be the checkbyte. During Multibuffer mode, following are the scenarios associated with the CC bit : - If CC bit is set during the reception of the data, then the byte that is received after the reception of the programmed no. of data bytes indicated by SCIFORMAT[18:16], is treated as a checksum byte. - If CC bit is set during the IDLE period (that is, during inter-frame space), then the next immediate byte will be treated as a checksum byte. A CE will immediately be flagged if there is a checksum error. This bit is automatically cleared once the checksum is successfully compared. Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Compare checksum on expected checkbyte
16	SC	R/W	0h	Send Checksum This mode is effective in LIN mode only. This bit is used by the transmitter with extended frames to send a checkbyte. In non-multibuffer mode the checkbyte will be sent after the current byte transmission. In multibuffer mode the checkbyte will be sent after the last byte count, indicated by the SCIFORMAT[18:16]). This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No checkbyte will be sent. 1h (R/W) = A checkbyte will be sent. This bit will automatically get cleared after the checkbyte is transmitted. The checksum will not be sent if this bit is set before transmitting the very first byte, that is, during interframe space.
15-9	RESERVED	R	0h	Reserved

Table 19-13. SCIGCR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	GENWU	R/W	0h	Generate wakeup signal. This bit controls the generation of a wakeup signal, by transmitting the TDO buffer value. This bit is cleared on reception of a valid sync break. Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Transmit TDO for wakeup. This bit will be cleared on a SWnRST (SCIGCR1.7)
7-1	RESERVED	R	0h	Reserved
0	POWERDOWN	R/W	0h	Power down. This bit is effective in LIN or SCI-compatible mode. When the powerdown bit is set, the SCI/LIN module attempts to enter local low-power mode. If the POWERDOWN bit is set while the receiver is actively receiving data and the wakeup interrupt is disabled, then the SCI/LIN will delay low-power mode from being entered until completion of reception. In LIN mode the user may set the POWERDOWN bit on Sleep Command reception or on idle bus detection (more than 4 seconds, that is, 80,000 cycles at 20 kHz) Reset type: SYSRSn 0h (R/W) = Normal operation 1h (R/W) = Request local low-power mode

19.7.2.4 SCISETINT Register (Offset = Ch) [reset = 0h]

The SCISETINT register is used to enable the various interrupts available in the LIN module.

Figure 19-29. SCISETINT Register

31	30	29	28	27	26	25	24
SETBEINT	SETPBEINT	SETCEINT	SETISFEINT	SETNREINT	SETFEINT	SETOEINT	SETPEINT
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		SETIDINT	RESERVED			SETRXINT	SETTXINT
R-0h		R/W1S-0h	R-0h			R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
SETTOA3WUSINT	SETTOAWUSINT	RESERVED	SETTIMEOUTINT	RESERVED		SETWAKEUPINT	SETBRKDTINT
R/W1S-0h	R/W1S-0h	R-0h	R/W1S-0h	R-0h		R/W1S-0h	R/W1S-0h

Table 19-14. SCISETINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SETBEINT	R/W1S	0h	Set bit error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when there is a bit error. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
30	SETPBEINT	R/W1S	0h	Set physical bus error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when a physical bus error occurs. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
29	SETCEINT	R/W1S	0h	Set checksum-error Interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when there is a checksum error. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
28	SETISFEINT	R/W1S	0h	Set inconsistent-sync-field-error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when there is an inconsistent sync field error. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.

Table 19-14. SCISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	SETNREINT	R/W1S	0h	Set no-response-error interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN module to generate an interrupt when a no-response error occurs. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
26	SETFEINT	R/W1S	0h	Set framing-error interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN module to generate an interrupt when a framing error occurs. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
25	SETOEINT	R/W1S	0h	Set overrun-error interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN module to generate an interrupt when an overrun error occurs. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
24	SETPEINT	R/W1S	0h	Set parity interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN module to generate an interrupt when a parity error occurs. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
23-14	RESERVED	R	0h	Reserved
13	SETIDINT	R/W1S	0h	Set Identification interrupt. This bit is effective in LIN mode only. This bit is set to enable interrupt once a valid matching identifier is received. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
12-10	RESERVED	R	0h	Reserved
9	SETRXINT	R/W1S	0h	Set Receiver interrupt. Setting this bit enables the SCI/LIN to generate a receive interrupt after a frame has been completely received and the data is being transferred from SCIRXSHF to SCIRD. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
8	SETTXINT	R/W1S	0h	Set Transmitter interrupt. Setting this bit enables the SCI/LIN to generate a transmit interrupt as data is being transferred from SCITD to SCITXSHF and the TXRDY bit is being set. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.

Table 19-14. SCISETINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	SETTOA3WUSINT	R/W1S	0h	Set Timeout After 3 Wakeup Signals interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN to generate an interrupt when there is a timeout after 3 wakeup signals have been sent. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
6	SETTOAWUSINT	R/W1S	0h	Set Timeout After Wakeup Signal interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN to generate an interrupt when there is a timeout after one wakeup signal has been sent. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
5	RESERVED	R	0h	Reserved
4	SETTIMEOUTINT	R/W1S	0h	Set timeout interrupt. This bit is effective in LIN mode only. Setting this bit enables the SCI/LIN to generate an interrupt when no LIN bus activity (bus idle) occurs for at least 4 seconds. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
3-2	RESERVED	R	0h	Reserved
1	SETWAKEUPINT	R/W1S	0h	Set wakeup interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit enables the SCI/LIN to generate a wakeup interrupt and thereby exit low-power mode. The wakeup interrupt is asserted on falling edge of the wakeup pulse. If enabled, the wakeup interrupt is asserted when local low-power mode is requested while the receiver is busy or if a low level is detected on the SCIRX pin during low-power mode. Wakeup interrupt is not asserted upon a wakeup pulse if the module is not in power down mode. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.
0	SETBRKDTINT	R/W1S	0h	Set break-detect interrupt. This bit is effective in SCI-compatible mode only. Setting this bit enables the SCI/LIN to generate an interrupt if a break condition is detected on the LINRX pin. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled.

19.7.2.5 SCICLEARINT Register (Offset = 10h) [reset = 0h]

The SCICLEARINT register is used to disable the enabled interrupts without accessing the SCISETINT register.

Figure 19-30. SCICLEARINT Register

31	30	29	28	27	26	25	24
CLRBEINT	CLRPBEINT	CLRCEINT	CLRISFEINT	CLRNREINT	CLRFEINT	CLROEINT	CLRPEINT
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		CLRIDINT	RESERVED			CLRRXINT	CLRTXINT
R-0h		R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
CLRTOA3WUSINT	CLRTOAWUSINT	RESERVED	CLRTIMEOUTINT	RESERVED		CLRWAKEUPINT	CLRBRKDTINT
R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R-0h		R/W1C-0h	R/W1C-0h

Table 19-15. SCICLEARINT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLRBEINT	R/W1C	0h	Clear Bit Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the bit error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
30	CLRPBEINT	R/W1C	0h	Clear Physical Bus Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the physical-bus error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
29	CLRCEINT	R/W1C	0h	Clear checksum-error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the checksum-error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
28	CLRISFEINT	R/W1C	0h	Clear Inconsistent-Sync-Field-Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the ISFE interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.

Table 19-15. SCICLEARINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	CLRNREINT	R/W1C	0h	Clear No-Reponse-Error Interrupt. This bit is effective in LIN mode only. Setting this bit disables the no-response error interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
26	CLRFEINT	R/W1C	0h	Clear Framing-Error Interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables framing-error interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
25	CLROEINT	R/W1C	0h	Clear Overrun-Error Interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the overrun interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
24	CLRPEINT	R/W1C	0h	Clear Parity Interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the parity error interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
23-14	RESERVED	R	0h	Reserved
13	CLRIDINT	R/W1C	0h	Clear Identifier interrupt. This bit is effective in LIN mode only. Setting this bit disables the ID interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
12-10	RESERVED	R	0h	Reserved
9	CLRRXINT	R/W1C	0h	Clear Receiver interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the receiver interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
8	CLRTXINT	R/W1C	0h	Clear Transmitter interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the transmitter interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.

Table 19-15. SCICLEARINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CLRTOA3WUSINT	R/W1C	0h	Clear Timeout After 3 Wakeup Signals interrupt. This bit is effective in LIN mode only. Setting this bit disables the timeout after 3 wakeup signals interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
6	CLRTOAWUSINT	R/W1C	0h	Clear Timeout After Wakeup Signal interrupt. This bit is effective in LIN mode only. Setting this bit disables the timeout after one wakeup signal interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
5	RESERVED	R	0h	Reserved
4	CLRTIMEOUTINT	R/W1C	0h	Clear Timeout interrupt. This bit is effective in LIN mode only. Setting this bit disables the timeout (LIN bus idle) interrupt. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
3-2	RESERVED	R	0h	Reserved
1	CLRWAKEUPINT	R/W1C	0h	Clear Wakeup interrupt. This bit is effective in LIN or SCI-compatible mode. Setting this bit disables the wakeup interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.
0	CLRBRKDTINT	R/W1C	0h	Clear Break-detect interrupt. This bit is effective in SCI-compatible mode only. Setting this bit disables the Break-detect interrupt. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt is disabled. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt is enabled. Writing a 1 to this bit will disable the interrupt and clear this bit.

19.7.2.6 SCISETINTLVL Register (Offset = 14h) [reset = 0h]

The SCISETINTLVL register is used to map individual interrupt sources to the INT1 interrupt line.

Figure 19-31. SCISETINTLVL Register

31	30	29	28	27	26	25	24
SETBEINT LVL	SETPBEINT LVL	SETCEINT LVL	SETISFEINT LVL	SETNREINT LVL	SETFEINT LVL	SETOEINT LVL	SETPEINT LVL
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		SETIDINTLVL	RESERVED			SETRXINTOVO	SETTXINTLVL
R-0h		R/W1S-0h	R-0h			R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
SETTOA3WU SINTLVL	SETTOAWU SINTLVL	RESERVED	SETTIMEOUT INTLVL	RESERVED		SETWAKEUP INTLVL	SETBRKDT INTLVL
R/W1S-0h	R/W1S-0h	R-0h	R/W1S-0h	R-0h		R/W1S-0h	R/W1S-0h

Table 19-16. SCISETINTLVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SETBEINTLVL	R/W1S	0h	Set Bit Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Bit Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
30	SETPBEINTLVL	R/W1S	0h	Set Physical Bus Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Physical Bus Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
29	SETCEINTLVL	R/W1S	0h	Set Checksum-error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Checksum-error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
28	SETISFEINTLVL	R/W1S	0h	Set Inconsistent-Sync-Field-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Inconsistent-Sync-Field-Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
27	SETNREINTLVL	R/W1S	0h	Set No-Reponse-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the No-Response-Error interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.

Table 19-16. SCISETINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	SETFEINTLVL	R/W1S	0h	Set Framing-Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Framing-Error interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
25	SETOEINTLVL	R/W1S	0h	Set Overrun-Error Interrupt Level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Overrun-Error interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
24	SETPEINTLVL	R/W1S	0h	Set Parity Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Parity error interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
23-14	RESERVED	R	0h	Reserved
13	SETIDINTLVL	R/W1S	0h	Set ID interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the ID interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
12-10	RESERVED	R	0h	Reserved
9	SETRXINTOVO	R/W1S	0h	Set Receiver interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the receiver interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
8	SETTXINTLVL	R/W1S	0h	Set Transmitter interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the transmitter interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
7	SETTOA3WUSINTLVL	R/W1S	0h	Set Timeout After 3 Wakeup Signals interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout after 3 wakeup signals interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
6	SETTOAWUSINTLVL	R/W1S	0h	Set Timeout After Wakeup Signal interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout after wakeup interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
5	RESERVED	R	0h	Reserved

Table 19-16. SCISETINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SETTIMEOUTINTLVL	R/W1S	0h	Set Timeout interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout interrupt level to the INT1 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
3-2	RESERVED	R	0h	Reserved
1	SETWAKEUPINTLVL	R/W1S	0h	Set Wakeup interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Wakeup interrupt level to the INT1 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.
0	SETBRKDTINTLVL	R/W1S	0h	Set Break-detect interrupt level. This bit is effective in SCI-compatible mode only. Writing to this bit maps the Break-detect interrupt level to the INT1 line. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line.

19.7.2.7 SCICLEARINTLVL Register (Offset = 18h) [reset = 0h]

The SCICLEARINTLVL register is used to map individual interrupt sources to the INT0 line.

Figure 19-32. SCICLEARINTLVL Register

31	30	29	28	27	26	25	24
CLRBEINTLVL	CLRPBEINTLVL	CLRCEINTLVL	CLRISFEINTLVL	CLRNREINTLVL	CLRFEINTLVL	CLROEINTLVL	CLRPEINTLVL
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		CLRIDINTLVL	RESERVED			CLRRXINTLVL	CLRTXINTLVL
R-0h		R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
CLRTOA3WUSINTLVL	CLRTOAWUSINTLVL	RESERVED	CLRTIMEOUTINTLVL	RESERVED		CLRWAKEUPINTLVL	CLRBRKDTINTLVL
R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R-0h		R/W1C-0h	R/W1C-0h

Table 19-17. SCICLEARINTLVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLRBEINTLVL	R/W1C	0h	Clear Bit Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Bit Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
30	CLRPBEINTLVL	R/W1C	0h	Clear Physical Bus Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Physical Bus Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
29	CLRCEINTLVL	R/W1C	0h	Clear Checksum-error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Checksum-error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
28	CLRISFEINTLVL	R/W1C	0h	Clear Inconsistent-Sync-Field-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the Inconsistent-Sync-Field-Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.

ADVANCE INFORMATION

Table 19-17. SCICLEARINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	CLRNREINTLVL	R/W1C	0h	Clear No-Reponse-Error interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the No-Response-Error interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
26	CLRFEINTLVL	R/W1C	0h	Clear Framing-Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Framing-Error interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
25	CLROEINTLVL	R/W1C	0h	Clear Overrun-Error Interrupt Level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Overrun-Error interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
24	CLRPEINTLVL	R/W1C	0h	Clear Parity Error interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Parity Error interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
23-14	RESERVED	R	0h	Reserved
13	CLRIDINTLVL	R/W1C	0h	Clear ID interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the ID interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
12-10	RESERVED	R	0h	Reserved
9	CLRRXINTLVL	R/W1C	0h	Clear Receiver interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the receiver interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
8	CLRTXINTLVL	R/W1C	0h	Clear Transmitter interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the transmitter interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.

Table 19-17. SCICLEARINTLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	CLRTOA3WUSINTLVL	R/W1C	0h	Clear Timeout After 3 Wakeup Signals interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout after 3 wakeup signals interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
6	CLRTOAWUSINTLVL	R/W1C	0h	Clear Timeout After Wakeup Signal interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout after wakeup interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
5	RESERVED	R	0h	Reserved
4	CLRTIMEOUTINTLVL	R/W1C	0h	Clear Timeout interrupt level. This bit is effective in LIN mode only. Writing to this bit maps the timeout interrupt level to the INT0 line. This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
3-2	RESERVED	R	0h	Reserved
1	CLRWAKEUPINTLVL	R/W1C	0h	Clear Wakeup interrupt level. This bit is effective in LIN or SCI-compatible mode. Writing to this bit maps the Wakeup interrupt level to the INT0 line. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. Writing a 0 to this bit has no effect. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.
0	CLRBRKDTINTLVL	R/W1C	0h	Clear Break-detect interrupt level. This bit is effective in SCI-compatible mode only. Writing to this bit maps the Break-detect interrupt level to the INT0 line. This field is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Interrupt level mapped to INT0 line. 1h (R/W) = Interrupt level mapped to INT1 line. Writing a 1 to this bit will map the interrupt to INT0 and clear this bit.

19.7.2.8 SCIFLR Register (Offset = 1Ch) [reset = 900h]

The SCIFLR register indicates the current status of the various interrupt sources of the LIN module.

Figure 19-33. SCIFLR Register

31	30	29	28	27	26	25	24
BE	PBE	CE	ISFE	NRE	FE	OE	PE
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	IDRXFLAG	IDTXFLAG	RXWAKE	TXEMPTY	TXWAKE	RXRDY	TXRDY
R-0h	R/W1C-0h	R/W1C-0h	R-0h	R-1h	R/W-0h	R/W1C-0h	R-1h
7	6	5	4	3	2	1	0
TOA3WUS	TOAWUS	RESERVED	TIMEOUT	BUSY	IDLE	WAKEUP	BRKDT
R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R-0h	R-0h	R/W1C-0h	R/W1C-0h

Table 19-18. SCIFLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	BE	R/W1C	0h	Bit Error Flag. This bit is effective in LIN mode only. This bit is set when there has been a bit error. This is detected by the bit monitor in the internal bit monitor. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No bit error detected. 1h (R/W) = Bit error detected.
30	PBE	R/W1C	0h	Physical Bus Error Flag. This bit is effective in LIN mode only. This bit is set when there has been a physical bus error. This is detected by the bit monitor in TED. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break Note: the PBE will only be flagged if no sync break can be generated. (because of a bus shortage to VBAT) or if no sync break delimiter can be generated (because of a bus shortage to GND). This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No physical bus error detected. 1h (R/W) = Physical bus error detected.

Table 19-18. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
29	CE	R/W1C	0h	Checksum Error Flag. This bit is effective in LIN mode only. This bit is set when there is checksum error detected by a receiving node. The type of checksum to be used depends on the SCIGCR1.CTYPE bit. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No Checksum error detected. 1h (R/W) = Checksum error detected.
28	ISFE	R/W1C	0h	Inconsistent Sync Field Error Flag. This bit is effective in LIN mode only. This bit is set when there has been an inconsistent Sync Field error detected by the synchronizer during header reception. See Section 19.3.1.5.2 for more information. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No Inconsistent Sync Field error detected. 1h (R/W) = Inconsistent Sync Field error detected.
27	NRE	R/W1C	0h	No-Response Error Flag. This bit is effective in LIN mode only. This bit is set when there is no response to a master's header completed within TFRAME_MAX. This timeout period is applied for message frames of unknown length (identifiers 0 to 61). This error is detected by the synchronizer of the module. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new sync break This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No No-Response error detected. 1h (R/W) = No-Response error detected.

Table 19-18. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	FE	R/W1C	0h	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 compatible 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 to generate an error interrupt if the RXERR INT ENA bit is set. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit - Reception of a new character (SCI-compatible mode), or frame (LIN mode) In multibuffer mode the frame is defined in the SCIFORMAT register. Reset type: SYSRSn 0h (R/W) = No framing error detected. 1h (R/W) = Framing error detected.
25	OE	R/W1C	0h	Overrun error flag. This bit is effective in LIN or SCI-compatible mode. This bit is set when the transfer of data from SCIRXSHF to SCIRD overwrites unread data already in SCIRD or the RDy buffers. Detection of an overrun error causes the LIN to generate an error interrupt if the SET OE INT bit is one. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit Reset type: SYSRSn 0h (R/W) = No overrun error detected. 1h (R/W) = Overrun error detected.
24	PE	R/W1C	0h	Parity error flag. This bit is effective in LIN or SCI-compatible mode. 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. For more information on parity checking, see the "SCI Global Control Register (SCIGCR1)" description. If the parity function is disabled (that is, SCIGCR1.2 = 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 =1. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Reception of a new character (SCI-compatible mode) or frame (LIN mode) - Writing a 1 to this bit Reset type: SYSRSn 0h (R/W) = No parity error or parity disabled. 1h (R/W) = Parity error detected.
23-15	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

Table 19-18. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	IDRXFLAG	R/W1C	0h	Identifier On Receive Flag. This bit is effective in LIN mode only. This flag is set once an identifier is received with an RX match and no ID-parity error. See Section 19.3.1.9 for more details. When this flag is set it indicates that a new valid identifier has been received on an RX match. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Reading the LINID register - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No valid ID received. 1h (R/W) = Valid ID RX received in LINID[23:16] on RX match.
13	IDTXFLAG	R/W1C	0h	Identifier On Transmit Flag. This bit is effective in LIN mode only. This flag is set once an identifier is received with a TX match and no ID-parity error. See Section 19.3.1.9 for more details. When this flag is set it indicates that a new valid identifier has been received on a TX match. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - RESET bit (SCIGCR0.0) - Setting SWnRESET - System reset - Reading the LINID register - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No valid ID received. 1h (R/W) = Valid ID received in LINID[23:16] on TX match.
12	RXWAKE	R	0h	Receiver wakeup detect flag. This bit is effective in SCI-compatible mode only. The SCI sets this bit to indicate that the data currently in SCIRD is an address. This bit is cleared by: - RESET bit - Setting the SWnRESET - System reset - Receipt of a data frame This bit is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = The data in SCIRD is not an address. 1h (R/W) = The data in SCIRD is an address. See Section 19.2.4.1 for more information on using the RXWAKE bit with sleep mode.

ADVANCE INFORMATION

Table 19-18. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	TXEMPTY	R	1h	Transmitter Empty flag. The value of this flag indicates the contents of the transmitter's buffer register(s) (SCITD/TDy) and shift register (SCITXSHF). In multibuffer mode, this flag indicates the value of the TDx registers and shift register (SCITXSHF). In non-multibuffer mode, this flag indicates the value of LINTDO (byte) and shift register (SCITXSHF). This bit is set by: - RESET bit (SCIGCR0.0) - Setting the SWnRESET (SCIGCR1.7 - System reset. Note: This bit does not cause an interrupt request. Reset type: SYSRSn 0h (R/W) = Compatible mode or LIN with no multibuffer: Transmitter buffer or shift register (or both) are loaded with data. In LIN mode using multibuffer mode: Multibuffer or shift register (or all) are loaded with data. 1h (R/W) = Compatible mode or LIN with no multibuffer: Transmitter buffer and shift registers are both empty. In LIN mode using multibuffer mode: Multibuffer and shift registers are all empty.
10	TXWAKE	R/W	0h	SCI transmitter wakeup method select. This bit is effective in SCI-compatible mode only. 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. TXWAKE is not cleared by the SWnRESET bit (SCIGCR1.7). Reset type: SYSRSn 0h (R/W) = Address-bit mode: Frame to be transmitted will be data (address bit = 0). Idle-line mode: Frame to be transmitted will be data. 1h (R/W) = Address-bit mode: Frame to be transmitted will be an address (address bit=1). Idle-line mode: 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).
9	RXRDY	R/W1C	0h	Receiver ready flag. In SCI compatibility mode, the receiver sets this bit to indicate that the SCIRD contains new data and is ready to be read by the CPU. In LIN mode, RXRDY is set once a valid frame is received in multibuffer mode, a valid frame being a message frame received with no errors. In non-multibuffer mode RXRDY is set for each received byte and will be set for the last byte of the frame if there are no errors. The SCI/LIN generates a receive interrupt when RXRDY flag bit is set if the interrupt-enable bit is set (SCISETINT.9). RXRDY is cleared by: - RESET bit (SCIGCR0.0) - Setting the SWnRESET - System reset - Writing a 1 to this bit - Reading SCIRD in while in SCI compatibility mode - Reading 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. Reset type: SYSRSn 0h (R/W) = No new data in SCIRD/RDy. 1h (R/W) = New data ready to be read from SCIRD.

Table 19-18. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	TXRDY	R	1h	Transmitter buffer register ready flag. When set, this bit indicates that the transmit buffer(s) register (SCITD in compatibility mode and LINTDO, LINTD1 in MBUF mode) is/are ready to get another character from a CPU write. In SCI compatibility mode, writing data to SCITD automatically clears this bit. In LIN mode, this bit is cleared once byte 0 (TD0) is written to LINTD0. This bit is set after the data of the TX buffer are shifted into the SCITXSHF register. . This bit is set to 1 by: - RESET bit (SCIGCR0.0) - Setting the SWnRESET (SCIGCR1.7) - System reset Note: The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0/1 register. Note: The transmit interrupt request can be eliminated until the next series of data is written into the transmit buffers LINTD0 and LINTD1, by disaLINg the corresponding interrupt via the SCICLEARINT register or by disaLINg the transmitter via the TXENA bit (SCIGCR1.25=0). Reset type: SYSRSn 0h (R/W) = Compatible mode: SCITD is full. LIN mode: The multibuffers are full. 1h (R/W) = Compatible mode: SCITD is ready to receive the next character. LIN mode: The multibuffers are ready to receive the next character(s).
7	TOA3WUS	R/W1C	0h	Timeout After 3 Wakeup Signals flag. This bit is effective in LIN mode only. This flag is set if there is no Sync Break received after 3 wakeup signals and a period of 1.5 seconds have passed. Such expiration time is used before issuing another round of wakeup signals. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No timeout after 3 wakeup signals. 1h (R/W) = Timeout after 3 wakeup signals and 1.5s time.
6	TOAWUS	R/W1C	0h	Timeout After Wakeup Signal flag. This bit is effective in LIN mode only. This bit is set if there is no Sync Break received after a wakeup signal has been sent. A minimum of 150 ms expiration time is used before issuing another wakeup signal. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No timeout after one wakeup signal (150 ms). 1h (R/W) = Timeout after one wakeup signal.
5	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

Table 19-18. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TIMEOUT	R/W1C	0h	LIN Bus IDLE timeout flag. This bit is effective in LIN mode only. This bit is set if there is no LIN bus activity for at least 4 seconds. LIN bus activity being a transition from recessive to dominant. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = No bus idle detected. 1h (R/W) = LIN bus idle detected.
3	BUSY	R	0h	Bus BUSY flag. This bit is effective in LIN mode and SCI-compatible mode. This 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 BUSY bit is cleared. If SET WAKEUP INT is set and power down is requested while this bit is set, the SCI/LIN automatically prevents low-power mode from being entered and generates wakeup interrupt. The BUSY bit is controlled directly by the SCI receiver but can be cleared by: - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset. Reset type: SYSRSn 0h (R/W) = Receiver is not currently receiving a frame. 1h (R/W) = Receiver is currently receiving a frame.
2	IDLE	R	0h	SCI receiver in idle state. This bit is effective in SCI-compatible mode only. 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 this state: - After a system reset - After a SCI software reset - After coming out of power down This bit is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = Idle period detected, the SCI is ready to receive. 1h (R/W) = Idle period not detected, the SCI will not receive any data.
1	WAKEUP	R/W1C	0h	Wakeup flag. This bit is effective in LIN mode only. This bit is set by the SCI/LIN 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.1) is set. This bit is cleared by: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register. - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - Writing a 1 to this bit This field is writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = Do not wake up from power-down mode. 1h (R/W) = Wake up from power-down mode.

Table 19-18. SCIFLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	BRKDT	R/W1C	0h	SCI break-detect flag. This bit is effective in SCI-compatible mode only. This bit is set when the SCI detects a break condition on the LINRX pin. A break condition occurs when the LINRX 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 BRKDT INT ENA bit is set. The BRKDT bit is cleared by the following: - Reading the corresponding interrupt offset in the SCIINTVECT0/1 register. - Setting the SWnRESET bit (SCIGCR1.7) - RESET bit (SCIGCR0.0) - System reset - By writing a 1 to this bit This bit is writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = No break condition detected. 1h (R/W) = Break condition detected.

19.7.2.9 SCIINTVECT0 Register (Offset = 20h) [reset = 0h]

The SCIINTVECT0 register indicates the offset for the INT0 interrupt line.

Figure 19-34. SCIINTVECT0 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											INTVECT0				
R-0h											R-0h				

Table 19-19. SCIINTVECT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	INTVECT0	R	0h	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 corresponding to the offset that was read. Note: The flags for the receive (SCIFLR.9) and the transmit (SCIFLR.8) interrupts cannot be cleared by reading the corresponding offset vector in this register (see detailed description in SCIFLR register). Reset type: SYSRSn

19.7.2.10 SCIINTVECT1 Register (Offset = 24h) [reset = 0h]

The SCIINTVECT1 register indicates the offset for the INT1 interrupt line.

Figure 19-35. SCIINTVECT1 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											INTVECT1				
R-0h											R-0h				

Table 19-20. SCIINTVECT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	INTVECT1	R	0h	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 corresponding to the offset that was read. Note: The flags for the receive (SCIFLR.9) and the transmit (SCIFLR.8) interrupts cannot be cleared by reading the corresponding offset vector in this register (see detailed description in SCIFLR register). Reset type: SYSRSn

19.7.2.11 SCIFORMAT Register (Offset = 28h) [reset = 0h]

The SCIFORMAT register is used to set up the character and frame lengths.

Figure 19-36. SCIFORMAT Register

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

ADVANCE INFORMATION

Table 19-21. SCIFORMAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-16	LENGTH	R/W	0h	Frame length control bits. In LIN mode, these bits indicate the number of bytes in the response field from 1 to 8 bytes. In buffered SCI mode, these bits indicate the number of characters. When these bits are used to indicate LIN response length (SCIGCR1[0] = 1), then when there is an ID RX match, this value should be updated with the expected length of the response. In buffered SCI mode, these bits indicate the number of characters with SCIFORMAT[2:0] bits per character, that is, these bits indicate the transmitter/receiver format for the number of characters: 1 to 8. There can be up to eight characters with eight bits each. Reset type: SYSRSn 0h (R/W) = The response field has 1 bytes/characters. 1h (R/W) = The response field has 2 bytes/characters. 2h (R/W) = The response field has 3 bytes/characters. 3h (R/W) = The response field has 4 bytes/characters. 4h (R/W) = The response field has 5 bytes/characters. 5h (R/W) = The response field has 6 bytes/characters. 6h (R/W) = The response field has 7 bytes/characters. 7h (R/W) = The response field has 8 bytes/characters.
15-3	RESERVED	R	0h	Reserved
2-0	CHAR	R/W	0h	Character length control bits. These bits are effective in SCI compatible mode only. These bits set the SCI character length from 1 to 8 bits. Note: In compatibility mode or buffered SCI mode, when data of fewer than eight bits in length is received, it is left justified in SCIRD/RDy and padded with trailing zeros. Data read from the SCIRD should be shifted by software to make the received data right justified. Note: Data written to the SCITD should be right justified but does not need to be padded with leading zeros. These bits are writable in SCI mode only. Reset type: SYSRSn 0h (R/W) = The character is 1 bits long. 1h (R/W) = The character is 2 bits long. 2h (R/W) = The character is 3 bits long. 3h (R/W) = The character is 4 bits long. 4h (R/W) = The character is 5 bits long. 5h (R/W) = The character is 6 bits long. 6h (R/W) = The character is 7 bits long. 7h (R/W) = The character is 8 bits long.

19.7.2.12 BRSR Register (Offset = 2Ch) [reset = 0h]

The BRSR register is used to configure the baud rate of the LIN module.

Figure 19-37. BRSR Register

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

Table 19-22. BRSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	M	R/W	0h	SCI/LIN 4-bit Fractional Divider Selection. (M) These bits are effective in LIN or SCI asynchronous mode. These bits are used to select a baud rate for the SCI/LIN module, and they are a fractional part for the baud rate specification. The M divider allows fine-tuning of the baud rate over the P prescaler with 15 additional intermediate values for each of the P integer values. Reset type: SYSRSn
23-16	SCI_LIN_PSH	R/W	0h	PRESCALER P (High Bits). SCI/LIN 24-bit Integer Prescaler Selection. These bits are used to select a baud rate for the SCI/LIN module. These bits are effective in LIN mode and SCI compatible mode. The SCI/LIN has an internally generated serial clock determined by the LIN module input clock and the prescalers P and M in this register. The SCI/LIN uses the 24-bit integer prescaler P value to select 1 of over 16,700,000 available baud rates. The additional 4-bit fractional prescaler M refines the baud rate selection. Reset type: SYSRSn
15-0	SCI_LIN_PSL	R/W	0h	PRESCALER P (Low Bits). SCI/LIN 24-bit Integer Prescaler Selection. These bits are used to select a baud rate for the SCI/LIN module. These bits are effective in LIN mode and SCI compatible mode. The SCI/LIN has an internally generated serial clock determined by the LIN module input clock and the prescalers P and M in this register. The SCI/LIN uses the 24-bit integer prescaler P value to select 1 of over 16,700,000 available baud rates. The additional 4-bit fractional prescaler M refines the baud rate selection. Reset type: SYSRSn

19.7.2.13 SCIED Register (Offset = 30h) [reset = 0h]

The SCIED register is a duplicate copy of SCIRD register that has no affect on the RXRDY flag for use with a debugger.

Figure 19-38. SCIED Register

31	30	29	28	27	26	25	24	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																								ED							
R-0h																								R-0h							

Table 19-23. SCIED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	ED	R	0h	Receiver Emulation Data. This bit is effective in SCI-compatible mode only. Reading SCIED(7-0) does not clear the RXRDY flag. This register should be used only by a debugger that must continually read the data buffer without affecting the RXRDY flag. Reset type: SYSRSn

19.7.2.14 SCIRD Register (Offset = 34h) [reset = 0h]

The SCIRD register is where received data is stored and can be read from.

Figure 19-39. SCIRD Register

31	30	29	28	27	26	25	24	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																								RD							
R-0h																								R-0h							

Table 19-24. SCIRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RD	R	0h	Received Data. This bit is effective in SCI-compatible mode only. 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 is set and a receive interrupt is generated if RX INT ENA (SCISETINT0.9) is set. When the data is read from SCIRD, the RXRDY flag is automatically cleared. 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, your software should perform a logical shift on the data by the correct number of positions to make it right justified. Reset type: SYSRSn

19.7.2.15 SCITD Register (Offset = 38h) [reset = 0h]

The SCITD register is where data to be transmitted is written to by application software.

Figure 19-40. SCITD Register

31	30	29	28	27	26	25	24	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																								TD							
R-0h																								R/W-0h							

Table 19-25. SCITD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	TD	R/W	0h	Transmit data This bit is effective in SCI-compatible mode only. Data to be transmitted is written to this register. The transfer of data from this register to the transmit shift register SCITXSHF sets the TXRDY flag (SCIFLR.23), which indicates that SCITD is ready to be loaded with another byte of data. Note: If TX INT ENA (SCISETINT.8) is set, this data transfer also causes an interrupt. Note: Data written to the SCIRD register that is fewer than eight bits long must be right justified, but it does not need to be padded with leading zeros. Reset type: SYSRSn

19.7.2.16 SCIPIO0 Register (Offset = 3Ch) [reset = 0h]

The SCIPIO0 register is used to enable the LINTX and LINRX pins.

Figure 19-41. SCIPIO0 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					TXFUNC	RXFUNC	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 19-26. SCIPIO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXFUNC	R/W	0h	Transmit pin function. This bit is effective in LIN or SCI mode. This bit defines the function of LINTX pin. Reset type: SYSRSn 0h (R/W) = LINTX pin is disabled. 1h (R/W) = LINTX pin is enabled.
1	RXFUNC	R/W	0h	Receive pin function. This bit is effective in LIN or SCI mode. This bit defines the function of the LINRX pin. Reset type: SYSRSn 0h (R/W) = LINRX pin is disabled. 1h (R/W) = LINRX pin is enabled.
0	RESERVED	R	0h	Reserved

19.7.2.17 SCIPIO1 Register (Offset = 40h) [reset = 0h]

The SCIPIO1 register determines the pin direction of the LINTX and LINRX pins.

Figure 19-42. SCIPIO1 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					TXDIR	RXDIR	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 19-27. SCIPIO1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXDIR	R/W	0h	Transmit pin direction. This bit is effective in LIN or SCI mode. This bit determines the data direction on the LINTX pin if it is configured with general-purpose I/O functionality (TX FUNC = 0). 0: general purpose input pin. 1: general-purpose output pin Reset type: SYSRSn
1	RXDIR	R/W	0h	Receive pin direction. This bit is effective in LIN or SCI mode. This bit determines the data direction on the LINRX pin if it is configured with general-purpose I/O functionality (RX FUNC = 0). 0: general purpose input pin. 1: general-purpose output pin Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

19.7.2.18 SCIPIO2 Register (Offset = 44h) [reset = 0h]

The SCIPIO2 register indicates the current status of the LINTX and LINRX pins.

Figure 19-43. SCIPIO2 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					TXIN	RXIN	RESERVED
R-0h					R-0h	R-0h	R-0h

Table 19-28. SCIPIO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TXIN	R	0h	Transmit data in. This bit is effective in LIN or SCI-compatible mode. This bit contains the current value on the LINTX pin. Reset type: SYSRSn
1	RXIN	R	0h	Receive data in. This bit is effective in LIN or SCI-compatible mode. This bit contains the current value on the LINRX pin. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

19.7.2.19 SCIPIO3 Register (Offset = 48h) [reset = 0h]

The SCIPIO3 register specifies the logic to be output on the LINTX and LINRX pins.

Figure 19-44. SCIPIO3 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					TXOUT	RXOUT	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 19-29. SCIPIO3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXOUT	R/W	0h	Transmit pin out. This bit is effective in LIN or SCI mode. This pin specifies the logic to be output on pin LINTX. Reset type: SYSRSn
1	RXOUT	R/W	0h	Receive pin out. This bit is effective in LIN or SCI mode. This pin specifies the logic to be output on pin LINRX. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

19.7.2.20 SCIPIO4 Register (Offset = 4Ch) [reset = 0h]

The SCIPIO4 register is used to set the logic output on the LINTX and LINRX pins.

Figure 19-45. SCIPIO4 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					TXSET	RXSET	RESERVED
R-0h					R/W1S-0h	R/W1S-0h	R-0h

Table 19-30. SCIPIO4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXSET	R/W1S	0h	Transmit pin set. This bit is effective in LIN or SCI mode. This bit sets the logic to be output on pin LINTX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit has no effect. 1h (R/W) = Writing a 1 to this bit sets the logic output on pin LINTX
1	RXSET	R/W1S	0h	Receive pin set. This bit is effective in LIN or SCI mode. This bit sets the logic to be output on pin LINRX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit has no effect. 1h (R/W) = Writing a 1 to this bit sets the logic output on pin LINRX.
0	RESERVED	R	0h	Reserved

19.7.2.21 SCIPIO5 Register (Offset = 50h) [reset = 0h]

The SCIPIO5 register is used to clear the logic output on the LINTX and LINRX pins.

Figure 19-46. SCIPIO5 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					TXCLR	RXCLR	RESERVED
R-0h					R/W1C-0h	R/W1C-0h	R-0h

Table 19-31. SCIPIO5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXCLR	R/W1C	0h	Transmit pin clear. This bit is effective in LIN or SCI mode. This bit clears the logic to be output on pin LINTX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit has no effect. 1h (R/W) = Writing a 1 to this bit clears the logic output on pin LINTX.
1	RXCLR	R/W1C	0h	Receive pin clear. This bit is effective in LIN or SCI mode. This bit clears the logic to be output on pin LINRX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit has no effect. 1h (R/W) = Writing a 1 to this bit clears the logic output on pin LINRX.
0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

19.7.2.22 SCIPIO6 Register (Offset = 54h) [reset = 0h]

The SCIPIO6 register is used to enable open-drain capability on the LINTX and LINRX pins.

Figure 19-47. SCIPIO6 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					TXPDR	RXPDR	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 19-32. SCIPIO6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXPDR	R/W	0h	Transmit pin open drain enable. This bit is effective in LIN or SCI mode. This bit enables open-drain capability in the output pin LINTX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit disables open-drain capability on output pin LINTX. 1h (R/W) = Writing a 1 to this bit enables open-drain capability on output pin LINTX.
1	RXPDR	R/W	0h	Receive pin open drain enable. This bit is effective in LIN or SCI mode. This bit enables open-drain capability in the output pin LINRX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit disables open-drain capability on output pin LINRX. 1h (R/W) = Writing a 1 to this bit enables open-drain capability on output pin LINRX.
0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

19.7.2.23 SCIPIO7 Register (Offset = 58h) [reset = 0h]

The SCIPIO7 register is used to disable pull control capability on the LINTX and LINRX pins.

Figure 19-48. SCIPIO7 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					TXPD	RXPD	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

Table 19-33. SCIPIO7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXPD	R/W	0h	Transmit pin pull control disable. This bit is effective in LIN or SCI mode. This bit disables pull control capability on the input pin LINTX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit enables pull control capability on input pin LINTX. 1h (R/W) = Writing a 1 to this bit disables pull control capability on input pin LINTX.
1	RXPD	R/W	0h	Receive pin pull control disable. This bit is effective in LIN or SCI mode. This bit disables pull control capability on the input pin LINRX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit enables pull control capability on input pin LINRX. 1h (R/W) = Writing a 1 to this bit disables pull control capability on input pin LINRX.
0	RESERVED	R	0h	Reserved

19.7.2.24 SCIPIO8 Register (Offset = 5Ch) [reset = 7h]

The SCIPIO8 register is used to select between pull types for the LINTX and LINRX pins.

Figure 19-49. SCIPIO8 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					TXPSL	RXPSL	RESERVED
R-0h					R/W-1h	R/W-1h	R-1h

Table 19-34. SCIPIO8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	TXPSL	R/W	1h	TX pin pull select. This bit is effective in LIN or SCI mode. This bit selects pull type in the input pin LINTX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit sets input pin LINTX to pull down. 1h (R/W) = Writing a 1 to this bit sets input pin LINTX to pull up.
1	RXPSL	R/W	1h	RX pin pull select. This bit is effective in LIN or SCI mode. This bit selects pull type in the input pin LINRX. Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit sets input pin LINRX to pull down. 1h (R/W) = Writing a 1 to this bit sets input pin LINRX to pull up.
0	RESERVED	R	1h	Reserved

19.7.2.25 LINCOMP Register (Offset = 60h) [reset = 0h]

The LINCOMPARE register is used to configure the sync delimiter and sync break extension.

Figure 19-50. LINCOMP 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						SDEL		RESERVED					SBREAK		
R-0h						R/W-0h		R-0h					R/W-0h		

Table 19-35. LINCOMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9-8	SDEL	R/W	0h	2-bit Sync Delimiter compare. These bits are effective in LIN mode only. These bits are used to configure the number of Tbit for the sync delimiter in the sync field. The time delay calculation for the synchronization delimiter is: TSDEL = (SDEL + 1)Tbit These bits are writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = The sync delimiter has 1 Tbit. 1h (R/W) = The sync delimiter has 2 Tbit. 2h (R/W) = The sync delimiter has 3 Tbit. 3h (R/W) = The sync delimiter has 4 Tbit.
7-3	RESERVED	R	0h	Reserved
2-0	SBREAK	R/W	0h	3-bit Sync Break extend. LIN mode only. These bits are used to configure the number of Tbits for the sync break to extend the minimum 13 Tbit in the Sync Field to a maximum of 20 Tbit. The time delay calculation for the sync break is: TSYNBRK = 13Tbit + SBREAK x Tbit These bits are writable in LIN mode only. Reset type: SYSRSn 0h (R/W) = The sync break has no additional Tbit. 1h (R/W) = The sync break has 1 additional Tbit. 2h (R/W) = The sync break has 2 additional Tbit. 3h (R/W) = The sync break has 3 additional Tbit. 4h (R/W) = The sync break has 4 additional Tbit. 5h (R/W) = The sync break has 5 additional Tbit. 6h (R/W) = The sync break has 6 additional Tbit. 7h (R/W) = The sync break has 7 additional Tbit.

19.7.2.26 LINRD0 Register (Offset = 64h) [reset = 0h]

The LINRD0 register contains the lower 4 bytes of the received LIN frame data.

Figure 19-51. LINRD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD0								RD1								RD2								RD3							
R-0h								R-0h								R-0h								R-0h							

Table 19-36. LINRD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RD0	R	0h	8-bit Receive Buffer 0 Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. A read of this byte clears the RXDY byte. Note: RD<x-1> is equivalent to Data byte <x> of the LIN frame. Reset type: SYSRSn
23-16	RD1	R	0h	8-bit Receive Buffer 1. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
15-8	RD2	R	0h	8-bit Receive Buffer 2. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
7-0	RD3	R	0h	8-bit Receive Buffer 3. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn

19.7.2.27 LINRD1 Register (Offset = 68h) [reset = 0h]

The LINRD1 register contains the upper 4 bytes of the received LIN frame data.

Figure 19-52. LINRD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD4								RD5								RD6								RD7							
R-0h								R-0h								R-0h								R-0h							

Table 19-37. LINRD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RD4	R	0h	8-bit Receive Buffer 4 Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
23-16	RD5	R	0h	8-bit Receive Buffer 5. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
15-8	RD6	R	0h	8-bit Receive Buffer 6. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn
7-0	RD7	R	0h	8-bit Receive Buffer 7. Each response data-byte that is received in the SCIRXSHFT register is transferred to the corresponding RDy register according to the number of bytes received. Reset type: SYSRSn

19.7.2.28 LINMASK Register (Offset = 6Ch) [reset = 0h]

The LINMASK register is used to configure the masks used for filtering incoming ID messages for receive and transmit frames.

Figure 19-53. LINMASK Register

31	30	29	28	27	26	25	24	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								RXIDMASK								RESERVED								TXIDMASK							
R-0h								R/W-0h								R-0h								R/W-0h							

Table 19-38. LINMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RXIDMASK	R/W	0h	Receive ID mask. This field is effective in LIN mode only.This 8-bit mask is used for filtering an incoming ID message and compare it to the ID-byte. A compare match of the received ID with the RX ID mask will set the ID RX flag and trigger and ID interrupt if enabled. A 0 bit in the mask indicates that bit is compared to the ID-byte. A 1 bit in the mask indicates that bit is filtered and therefore not used in the compare. When HGENCTRL is set to 1, this field must be set to 0xFF. Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-0	TXIDMASK	R/W	0h	Transmit ID mask. This field is effective in LIN mode only. This 8-bit mask is used for filtering an incoming ID message and compare it to the ID-byte. A compare match of the received ID with the TX ID Mask will set the ID TX flag and trigger an ID interrupt if enabled. A 0 bit in the mask indicates that bit is compared to the ID-byte. A 1 bit in the mask indicates that bit is filtered and therefore not used for the compare. When HGENCTRL is set to 1, this field must be set to 0xFF. Reset type: SYSRSn

19.7.2.29 LINID Register (Offset = 70h) [reset = 0h]

The LINID register contains the identification fields for LIN communication.
NOTE: For software compatibility with future LIN modules, the HGEN CTRL bit must be set to 1, the RX ID MASK field must be set to FFh, and the TX ID MASK field must be set to FFh.

Figure 19-54. LINID Register

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

Table 19-39. LINID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	RECEIVEDID	R	0h	Received ID. This bit is effective in LIN mode only. This byte contains the current message identifier. During header reception the received ID is copied from the SCIRXSHF register to this byte if there is no ID-parity error and there has been an RX/TX match. Note: If a framing error (FE) is detected during ID reception, the received ID will also not be copied to the LINID register. Reset type: SYSRSn
15-8	IDSLAVETASKBYTE	R/W	0h	ID Slave Task byte. This field is effective in LIN mode only. This byte contains the identifier to which the received ID of an incoming header will be compared in order to decide whether a RX response, a TX response, or no action needs to be done by the LIN node. These bits are writable in LIN mode only. Reset type: SYSRSn
7-0	IDBYTE	R/W	0h	ID byte. This field is effective in LIN mode only. This byte is the LIN mode message ID. On a master node, a write to this register by the CPU initiates a header transmission. For a slave task, this byte is used for message filtering when HGENCTRL (SCIGCR1.12) is '0'. These bits are writable in LIN mode only. Reset type: SYSRSn

19.7.2.30 LINTD0 Register (Offset = 74h) [reset = 0h]

The LINTD0 register contains the lower 4 bytes of the data to be transmitted.
NOTE: TD<x-1> is equivalent to Data byte <x> of the LIN frame.

Figure 19-55. LINTD0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD0								TD1								TD2								TD3							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 19-40. LINTD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TD0	R/W	0h	8-bit Transmit Buffer 0. Byte 0 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Once byte 0 is written in TDO buffer, transmission will be initiated. Reset type: SYSRSn
23-16	TD1	R/W	0h	8-bit Transmit Buffer 3. Byte 1 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
15-8	TD2	R/W	0h	8-bit Transmit Buffer 2. Byte 2 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
7-0	TD3	R/W	0h	8-bit Transmit Buffer 3. Byte 3 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn

19.7.2.31 LINTD1 Register (Offset = 78h) [reset = 0h]

The LINTD1 register contains the upper 4 bytes of the data to be transmitted.
NOTE: TD<x-1> is equivalent to Data byte <x> of the LIN frame.

Figure 19-56. LINTD1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD4								TD5								TD6								TD7							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 19-41. LINTD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TD4	R/W	0h	8-bit Transmit Buffer 4. Byte4 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
23-16	TD5	R/W	0h	8-bit Transmit Buffer 5. Byte 5 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
15-8	TD6	R/W	0h	8-bit Transmit Buffer 6. Byte 6 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn
7-0	TD7	R/W	0h	8-bit Transmit Buffer 7. Byte 7 to be transmitted is written into this register and then copied to SCITXSHF for transmission. Reset type: SYSRSn

19.7.2.32 MBRSR Register (Offset = 7Ch) [reset = 5DCh]

The MBRSR register is used to configure the expected maximum baud rate of the LIN network.

Figure 19-57. MBRSR Register

31	30	29	28	27	26	25	24	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																			MBR												
R-0h																			R/W-5DCh											

Table 19-42. MBRSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12-0	MBR	R/W	5DCh	Maximum Baud Rate Prescaler. This field is effective in LIN mode only. This 13-bit prescaler is used during the synchronization phase (see Section 19.3.1.5.2) of a slave module if the ADAPT bit is set. In this way, a SCI/LIN slave using an automatic or select bit rate modes detects any LIN bus legal rate automatically. The MBR value should be programmed to allow a maximum baud rate that is not more than 10% above the expected operating baud rate in the LIN network. Otherwise a s 0x00 data byte could mistakenly be detected as sync break. The default value is for a 30 MHz LINCLK (0x5DC). This MBR prescaler is used by the wakeup and idle time counters for a constant expiration time relative to a 20 kHz rate. Reset type: SYSRSn

19.7.2.33 IODFTCTRL Register (Offset = 90h) [reset = 0h]

The IODFTCTRL register is used to emulate various error and test conditions.

Figure 19-58. IODFTCTRL Register

31	30	29	28	27	26	25	24
BERRENA	PBERRENA	CERRENA	ISFERRENA	RESERVED	FERRENA	PERRENA	BRKDTERRENA
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED			PINSAMPLEMASK		TXSHIFT		
R/W-0h			R/W-0h		R/W-0h		
15	14	13	12	11	10	9	8
RESERVED				IODFTENA			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED						LPBENA	RXPENA
R-0h						R/W-0h	R/W-0h

Table 19-43. IODFTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	BERRENA	R/W	0h	Bit Error Enable bit. This bit is effective in LIN mode only. This bit is used to create a Bit Error. When this bit is set, the bit received is ORed with 1 and passed to the Bit monitor circuitry. Reset type: SYSRSn
30	PBERRENA	R/W	0h	Physical Bus Error Enable bit. This bit is effective in LIN mode only. This bit is used to create a Physical Bus Error. When this bit is set, the bit received during Sync Break field transmission is ORed with 1 and passed to the Bit monitor circuitry Reset type: SYSRSn
29	CERRENA	R/W	0h	Checksum Error Enable bit. This bit is effective in LIN mode only. This bit is used to create a Checksum Error. When this bit is set, the polarity of the CTYPE (checksum type) in the receive checksum calculator is changed so that a checksum error is generated. Reset type: SYSRSn
28	ISFERRENA	R/W	0h	Inconsistent Sync Field Error Enable bit. This bit is effective in LIN mode only. This bit is used to create an ISF error. When this bit is set, the bit widths in the sync field are varied so that the ISF check fails and the error flag is set. Reset type: SYSRSn
27	RESERVED	R	0h	Reserved
26	FERRENA	R/W	0h	Frame Error Enable Bit. This bit is used to create a Frame Error. When this bit is set, the stop bit received is ANDed with '0' and passed to the stop bit check circuitry. Reset type: SYSRSn
25	PERRENA	R/W	0h	Parity Error Enable bit. This bit is effective in SCI-compatible mode only. This bit is used to create a Parity Error. When this bit is set, in SCI-compatible mode, the parity bit received is toggled so that a parity error occurs. Reset type: SYSRSn

ADVANCE INFORMATION

Table 19-43. IODFTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	BRKDTERRENA	R/W	0h	Break Detect Error Enable bit. This bit is effective in SCI-compatible mode only. This bit is used to create BRKDT error (SCI mode only). When this bit is set, 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 Tbits so that a BRKDT error occurs. Reset type: SYSRSn
23-21	RESERVED	R/W	0h	Reserved
20-19	PINSAMPLEMASK	R/W	0h	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 correctly with the majority detection circuitry. Note: During IODFT mode testing for the pin sample mask, the prescalar P must be programmed to be greater than 2. Reset type: SYSRSn 0h (R/W) = No Mask 1h (R/W) = Invert the TX Pin value at TBIT_CENTER 2h (R/W) = Invert the TX Pin value at TBIT_CENTER + SCLK 3h (R/W) = Invert the TX Pin value at TBIT_CENTER + 2 SCLK
18-16	TXSHIFT	R/W	0h	Transmit shift. These bits define the delay by which the value on LINTX is delayed so that the value on LINRX is asynchronous. (Not applicable to Start Bit) Reset type: SYSRSn 0h (R/W) = No Delay 1h (R/W) = Delay by 1 SCLK 2h (R/W) = Delay by 2 SCLK 3h (R/W) = Delay by 3 SCLK 4h (R/W) = Delay by 4 SCLK 5h (R/W) = Delay by 5 SCLK 6h (R/W) = Delay by 6 SCLK 7h (R/W) = Delay by 7 SCLK
15-12	RESERVED	R	0h	Reserved
11-8	IODFTENA	R/W	0h	IO DFT Enable Key This field is used to enable the IODFT mode of the SCI/LIN module for testing. Reset type: SYSRSn 0h (R/W) = IODFT is disabled 1h (R/W) = IODFT is disabled 2h (R/W) = IODFT is disabled 3h (R/W) = IODFT is disabled 4h (R/W) = IODFT is disabled 5h (R/W) = IODFT is disabled 6h (R/W) = IODFT is disabled 7h (R/W) = IODFT is disabled 8h (R/W) = IODFT is disabled 9h (R/W) = IODFT is disabled Ah (R/W) = IODFT is enabled Bh (R/W) = IODFT is disabled Ch (R/W) = IODFT is disabled Dh (R/W) = IODFT is disabled Eh (R/W) = IODFT is disabled Fh (R/W) = IODFT is disabled
7-2	RESERVED	R	0h	Reserved

Table 19-43. IODFTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	LPBENA	R/W	0h	Module loopback enable. 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. Reset type: SYSRSn 0h (R/W) = Digital loopback is enabled. 1h (R/W) = Analog loopback is enabled in module I/O DFT mode (when IODFTENA = 1010)
0	RXPENA	R/W	0h	Module Analog loopback through receive pin enable. 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 only. Reset type: SYSRSn 0h (R/W) = Analog loopback through the transmit pin is enabled. 1h (R/W) = Analog loopback through the receive pin is enabled.

20 Radar Data InterFace

20.1 Features

20.1.1 Supported

- Chirp mode – Data is sent based on the Chirp Size
  - This module does not have any output signal to mark the chirp boundaries.
  - At the SoC level the dfe_chirp_start could be multiplexed to sent out or it needs to be handled in software.
  - Sideband data can be sent in this mode.
  - Supports delayed start of the data capture. DDR clock is started first. Data capture at the next chirp bound is supported.
- CW CZ mode support (benefit is for CZ/sysval lab activities)
  - Sideband data cannot be sent in this mode
  - Supports delayed start of the data capture. DDR clock is started first. Data capture can be started later. The output be aligned such that the data aligned with the first frame_clk is deterministic.
- Support for CRC-16
  - CRC polynomial used - X^**16** + X^15 + X^2 +1
  - CRC is calculated for chirp data + sideband data.
  - Reset value of the polynomial = 0xFFFF
- Scrambling

Use cases support

• Finite chirping with following data formats: – ADC_ONLY – CP_ADC_CQ (ADC data + Chirp Index + ADC Saturation Counts)	Supported
• Infinite chirping capture (while device is already chirping) with following data format: – CP_ADC (ADC data + Chirp Index)	Not Supported
• Continuous streaming capture (while device is already streaming) with following data format: – ADC_ONLY	Supported

20.1.2 Limitations

- Sync pattern every 1K samples - is not supported.
  - to enable starting of sampling by external receiver at any random time (not supported)
- In delayed start, there is no support for starting the frame on a next frame start. This will have to managed by software.
- To support sideband data, it is not possible to support partial sideband data. Either all the sideband data are supported or none. Saturation Count output from DFE to be necessarily enabled when sideband data is enabled for RDIF.

20.2 Key Spec Limits

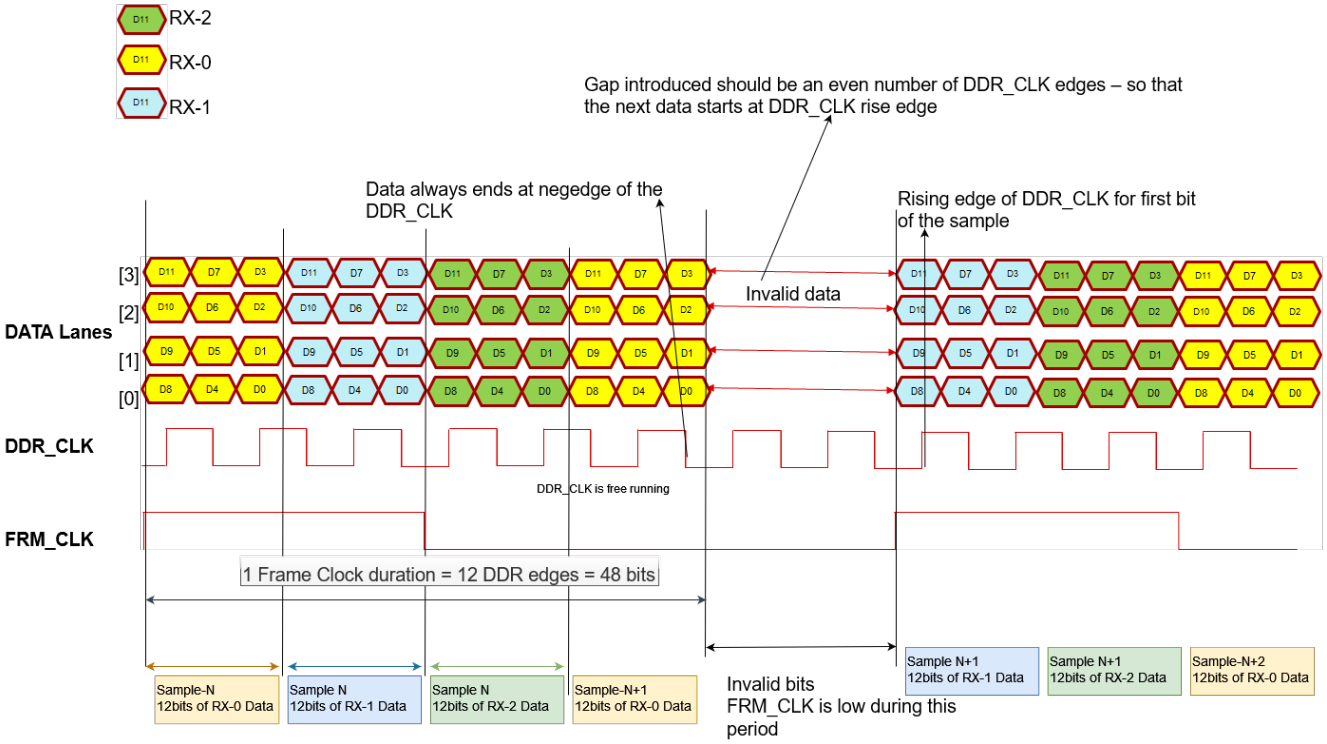
Parameter	Description	Min Limit	Max Limit
DFE Clock	Clock from DFE	-	12.5MHz
Sample Width	Number of bits per sample – DFE supports 12b/14b/16b	12b	12b
Number of Samples captured per Chirp (non-CW mode)	Number of samples per chirp per channel. **This should be a multiple of 4.**	16 samples – for single channels 16 samples (16*2) – for two channels 16 samples (16*3)– for three channels	65535 – for single channel 65535x2 – for two channels 65535x3 – for three channels

Parameter	Description	Min Limit	Max Limit
Serializer Clock	Clock used to generate the DDR clock and serialize the data	DFE_CLK x 8 (100MHz)	400MHz – If LVDS was used.

20.3 Usage Model and Assumptions

S No	Assumptions
1	4x CMOS LANE Interface – DATA[3:0] and FRM_CLK and DDR_CLK
2	Interface would support only 12-bit data mode .
3	Typically, the data rate is chosen higher than the DFE output data rate. In case there is not sufficient data in the FIFO (DFE being slower than RDIF output), the FRM CLK signal shall be gated.
4	Data rate would always be such that skew between the DFE output (including CHKSUM, monitoring and sync data) and output data rate does not cause the Data FIFO overflow. In case DFE output data rate is higher than output data rate, max acceptable skew is 12.5% (Typical case of 12.5MSPS DFE output and 400MHz output data rate) for max duration of 1024 samples.
5	CHKSUM, Monitoring (Max Freq Error, Saturation Count) and Receiver sync (Frame number, chirp number) data is always appended at the tail of transmitted data packet in a ‘single’ defined manner. This is achieved using a simple scheme by counting number of output samples to check the data packet boundary. Data packet is here referred to samples in a chirp. It is assumed that the Monitoring data would be available along with the last DFE samples output and hence before the LVDS interface switches mux to transmit that data.
6	Monitoring and Receiver Sync data is implemented as Ping-pong registers.
7	For lesser than 3 RX’s enabled use-case, the lower RX number is mapped to lower CH number during the data packing and data on data lanes. For ex: If RX2 and RX3 are enabled, RX2 is mapped to CH1 and RX3 is mapped to CH2 during data packing and hence for data lanes output.
8	As there can be jitter introduced in the DFE output, We have to assume 2 samples in one 12.5MSPS duration from the DFE. Data latching in the FIFO shall be at least 3x of the worst case DFE output. i.e 12.5x2x3 = 75MHz.
9	Data Write and Read from the FIFO shall be implemented in sample by sample manner rather than Memory row manner. This would keep the implementation simple.
10	In chirp accumulation mode of DFE, the final accumulated chirp data will be sent from DFE to RDIF. Therefore in chirp accumulation mode, the raw DFE data will not be accessible through RDIF, only the final accumulated chirp data can be read through RDIF.

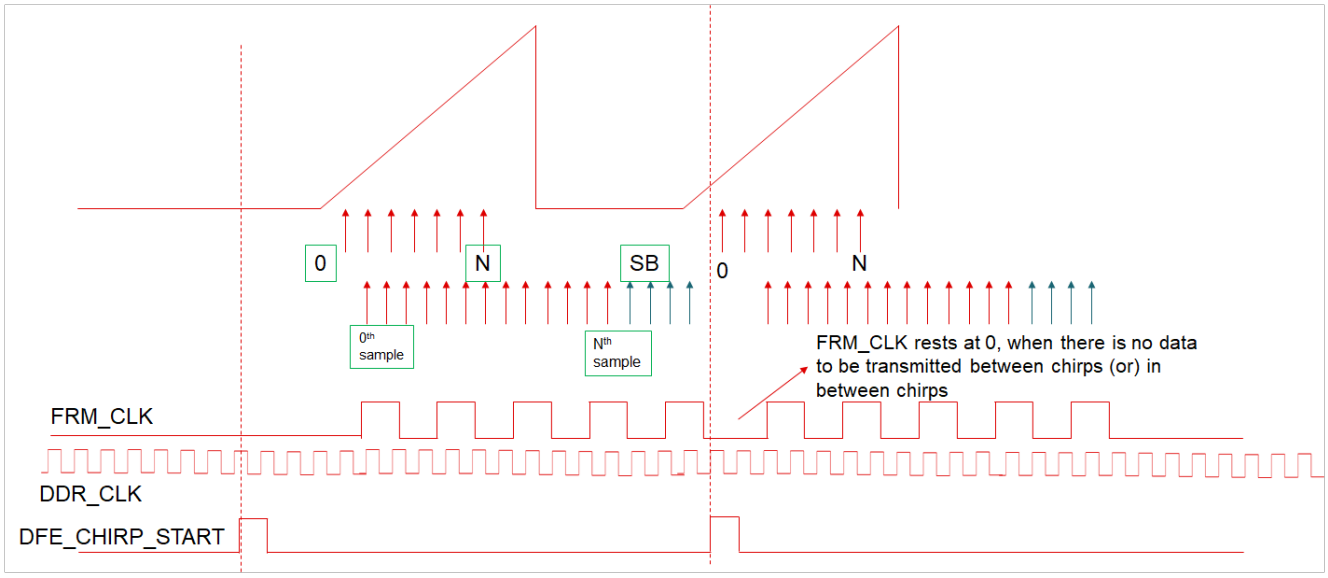
20.4 Data Format



The following are the data format requirements which are shown in the diagram above and the excel sheet below.

- The samples are sent one channel by one channel as shown in the diagram above. All the 12-bits of one channel are sent on 4 data lanes in 3 DDR_CLK edges, followed by next RX channel.
- The frame clock (FRM_CLK) spans 12 DDR_CLK edges and 48 bits are sent in 1 FRM_CLK
- The FRM_CLK can have gaps in between. This is required as the interface rate is greater than the incoming rate
- DDR_CLK is continuous.
- DDR_CLK is generated from 400MHz ADC CLK (one of the ADC CLKs) - selected for the DFE. It is the same 400MHz clock selected for DFE.
- New sample always starts at the rise edge of the DDR_CLK
- The FRM_CLK is valid for the entire data bit and is meets the Tsu/Th wrt DDR_CLK.

Figure 20-1. RDIF Data Format



BOUNDARY CONDITIONS:

1.The last sample of "Nth Chirp" has to be sent out on the serializer before next DFE_CHIRP_START rise edge for the "N+1th Chirp".

20.5 Sideband Data Format

The side band data consists of the following:

1. FRAME_CNT[11:0] - LSB 12 bits of this count
2. BURST_CNT[11:0] - LSB 12 bits of this count
3. CHIRP_CNT[11:0] - LSB 12 bits of this count
4. SATURATION_CNT[35:0] - 12b per channel (per channel, within each 12bits samples, <7:0> is valid, and the MSB 4bits are 0.
   a. There are three fixed slots named SATURATION_CNT_CH3, SATURATION_CNT_CH2 and SATURATION_CNT_CH1 - each 12b samples.
   b.

Channels Enabled	Slot SATURATION_CNT_CH3	Slot SATURATION_CNT_CH2	Slot SATURATION_CNT_CH1
All three channels enabled	DFE Channel 3	DFE Channel 2	DFE Channel 1
Two Channels Enabled #1,2	DFE Channel 2	DFE Channel 1	All zeros
Two Channels Enabled #2,3	DFE Channel 3	DFE Channel 2	All zeros
Two Channels Enabled #1,3	DFE Channel 3	DFE Channel 1	All zeros
One channel Enabled #1	DFE Channel 1	All zeros	All zeros
One channel Enabled #2	DFE Channel 2	All zeros	All zeros
One channel Enabled #3	DFE Channel 3	All zeros	All zeros

5. CRC-16[23:0] - MSB 8 bits are padded with zeros. (CRC is calculated for Chirp data + Sideband data)

The diagram below is shown for CFG_MSB_LSB_OPTIONS = "10".

Side Band Data	Side Band Data Generation Module	Sample at Event
Frame_cnt	Frame Timer	DFE Chirp Start (RX_DFE_START)

ADVANCE INFORMATION

Side Band Data	Side Band Data Generation Module	Sample at Event
Burst Cnt	Timing Engine	DFE Chirp Start (RX_DFE_START)
Chirp Cnt	Timing Engine	DFE Chirp Start (RX_DFE_START)
Saturation Count	DFE	DFE Chirp End (RX_SAT_CNT_VALID)

20.6 CW CZ MODE

Continuous Wave Characterization (CW CZ) Mode.

- In this mode of operation, the data will be streamed to output as when it arrives. 72bits of data (12 samples) are collected and then sent to output in three frame clocks.
  - Single Channel - 12 samples (two rows in the memory). This can be used when only one channel has to characterized. The user can configure the channel to be selected.
  - Two Channels - 6 samples per channel and that becomes 12 samples total (two rows in the memory). This can be used when two channels has to characterized. The user can configure the channel to be selected.
  - Three Channels - 4 samples per channel and that becomes 12 samples total (two rows in the memory). This can be used when all three channels has to characterized, but the max ADC sampling rate will be limited (due to 400Mbps limit vs. 450Mbps requirement).
  - Illustrated in the diagram below.
  - After sending 12 samples, if there is another 12 samples in the memory, the transmission will be started, else the FRM_CLK will be set to LOW and the data transmission will stop at the output (all -zeros) until the next 12 samples are available.
- Sideband data is not transmitted in this mode.
- In the CW CZ mode, the max ADC sampling rate will be limited (due to 400Mbps limit vs. 450Mbps requirement).

Max Sampling Rate

- In the CW CZ mode, the max ADC sampling rate will be limited (due to 400Mbps limit vs. 450Mbps requirement) for three channel case.

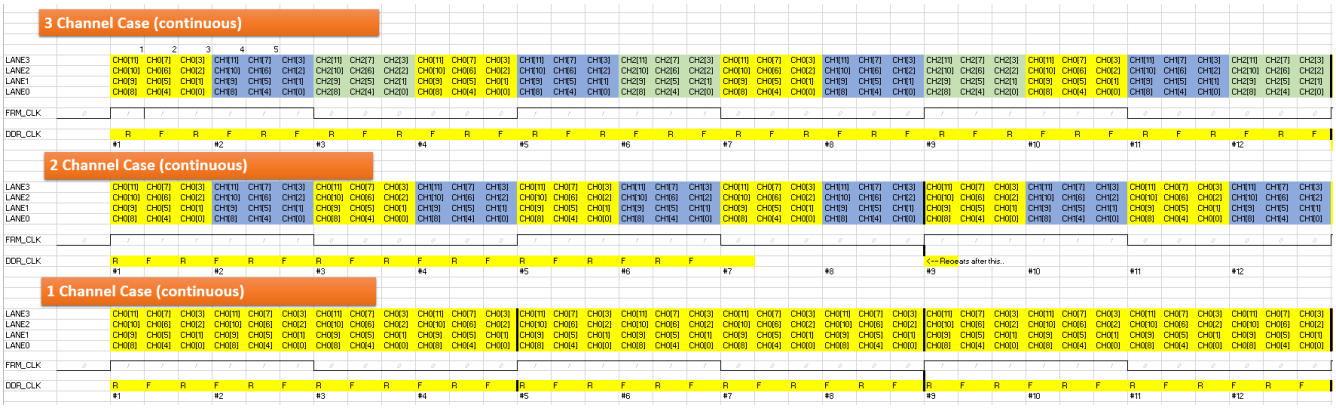


Figure 20-2. CW CZ Mode Example for CFG_MSB_LSB_OPTIONS = "10"

The diagram above is shown for cfg_msb_lsb_options = "10".

20.6.1 CW Delayed Start - Frame Alignment

Channel Configuration	Data Format (First Data Sample on the First frame_clk edge) after the data capture is enabled
1 Channel	Selected channel. If RX0 is selected, then the data is RX0. If RX1 is selected, then the data is RX1. If RX2 is selected, then the data is RX2.
2 channel - RX0 & RX1	RX0 is the first data
2 channel - RX0 & RX2	RX0 is the first data

Channel Configuration	Data Format (First Data Sample on the First frame_clk edge) after the data capture is enabled
2 channel - RX1 & RX2	RX1 is the first data
3 channel	Rx0 data is the first data

20.7 High Level Block Diagram

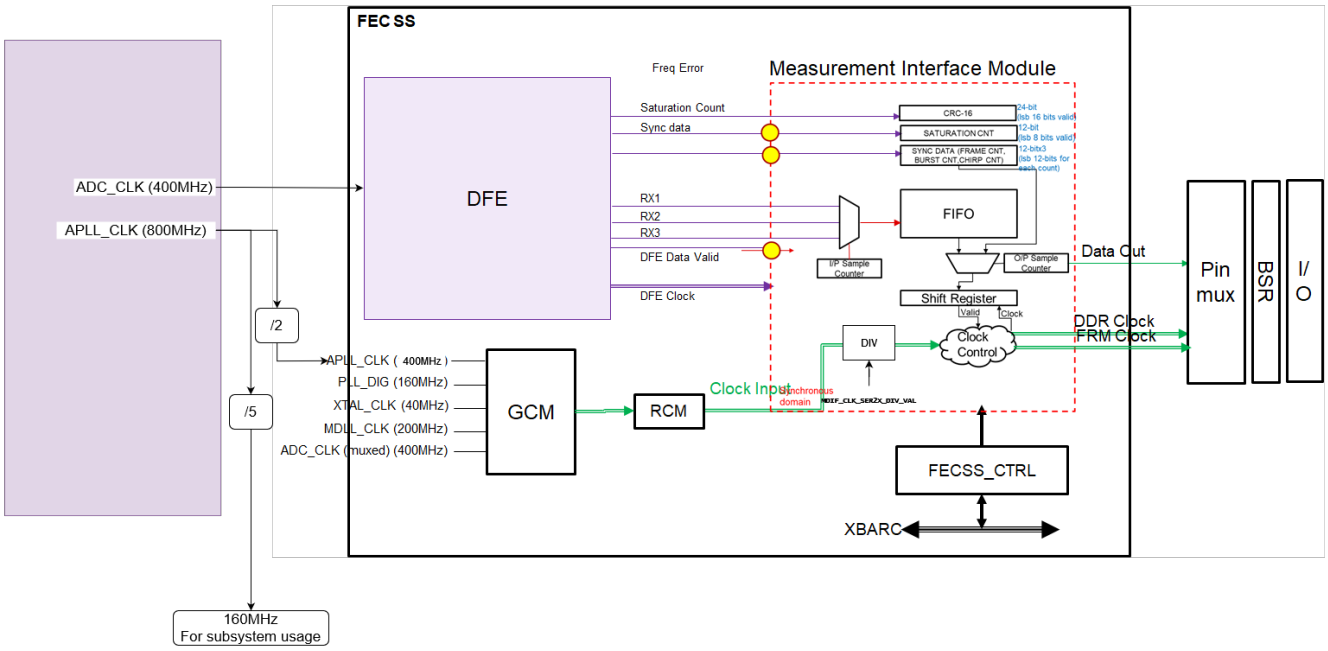


Figure 20-3. High Level Block Diagram

20.8 Microarchitecture

This module has the following sub-blocks:

- Input Formatter
- Arbiter
- Memory
- Output Buffer
- CRC Calc Engine
- Shift Register
- Reset & Clock Module

20.8.1 Input Formatter - Channel Selection

64xx has three ADC and the corresponding output from DFE is rx1_dat, rx2_dat, rx3_dat. Using RDIF, it is possible to capture any one or two or all three of the DFE output. The configuration bit used program which channel is enabled is cfg_rx1/2/3_en

This table describes how the data is stored in the input buffer and the memory for various use cases of cfg_rx1/2/3_en. The diagrams at the bottom of the slide, illustrates the data packing format.

In the memory, the data is packed without any gap. DFE_DATA_VALID continuously HIGH is the highest possible throughput scenario. Under this scenario, consider the single channel, two channel and three channel cases.

The memory width is chosen to hold 6 samples – LCM of 1 (one sample per cycle/Single channel), 2 (two sample per cycle / two channels enabled) and 3 (three samples per cycle / three channels enabled).

- –Single Channel : 3 samples are accumulated to for 36b data, then assert wr_en to memory. Hence there will be one request every 3 cycles.

- –Two Channel : In every cycle, two samples arrive, they will be stored in the input buffer as 24b and then assert wr_en to memory. (MSB 12b [35:24] remain unused).
- –Three Channel : In every cycle, three samples arrive, they will be stored in the input buffer as 36b and then assert wr_en to memory.

The input mux before each 12b buffer will choose which channel will be written to the input buffer.

cfg_ rx1_ en	cfg_ rx2_ en	cfg_ rx3_ en	Description	Stored in Memory	Stored in Input Buffer	Write to memory in "N" DFE_CLK cycles
1	0	0	Single Channel – RX1	One RX Use Case (See below diagram)		Once every 3 cycles.
0	1	0	Single Channel – RX2	One RX Use Case (See below diagram)		Once every 3 cycles.
0	0	1	Single Channel – RX3	One RX Use Case (See below diagram)		Once every 3 cycles.
1	1	0	Two Channels – RX1/2	Two RX Use Case (See below diagram)		Every Cycle
0	1	1	Two Channels – RX2/3	Two RX Use Case (See below diagram)		Every Cycle
1	0	1	Two Channels – RX1/3	Two RX Use Case (See below diagram)		Every Cycle
1	1	1	Three Channels – RX1/2/3	Three RX Use Case (See below diagram)		Every Cycle

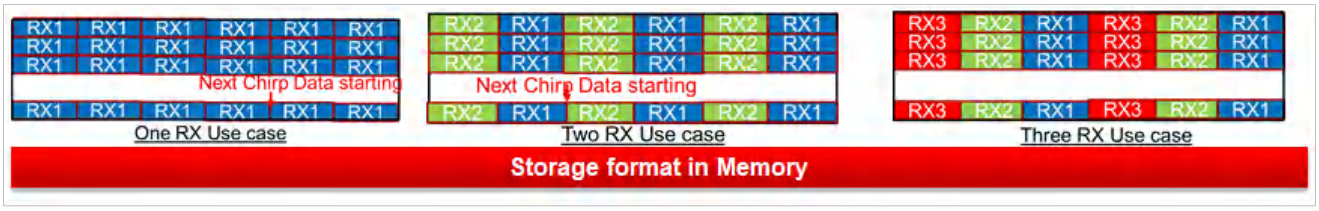


Figure 20-4. RDIF Input Formatter 1

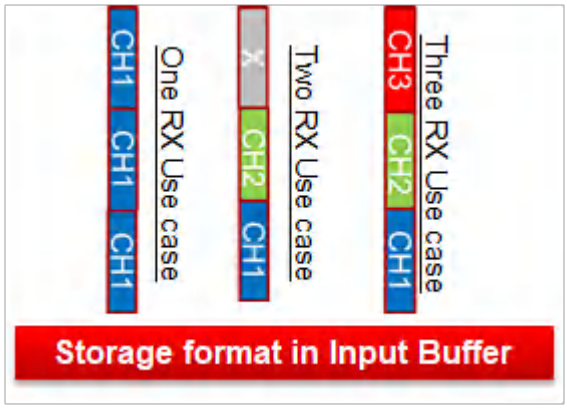


Figure 20-5. RDIF Input Formatter 2

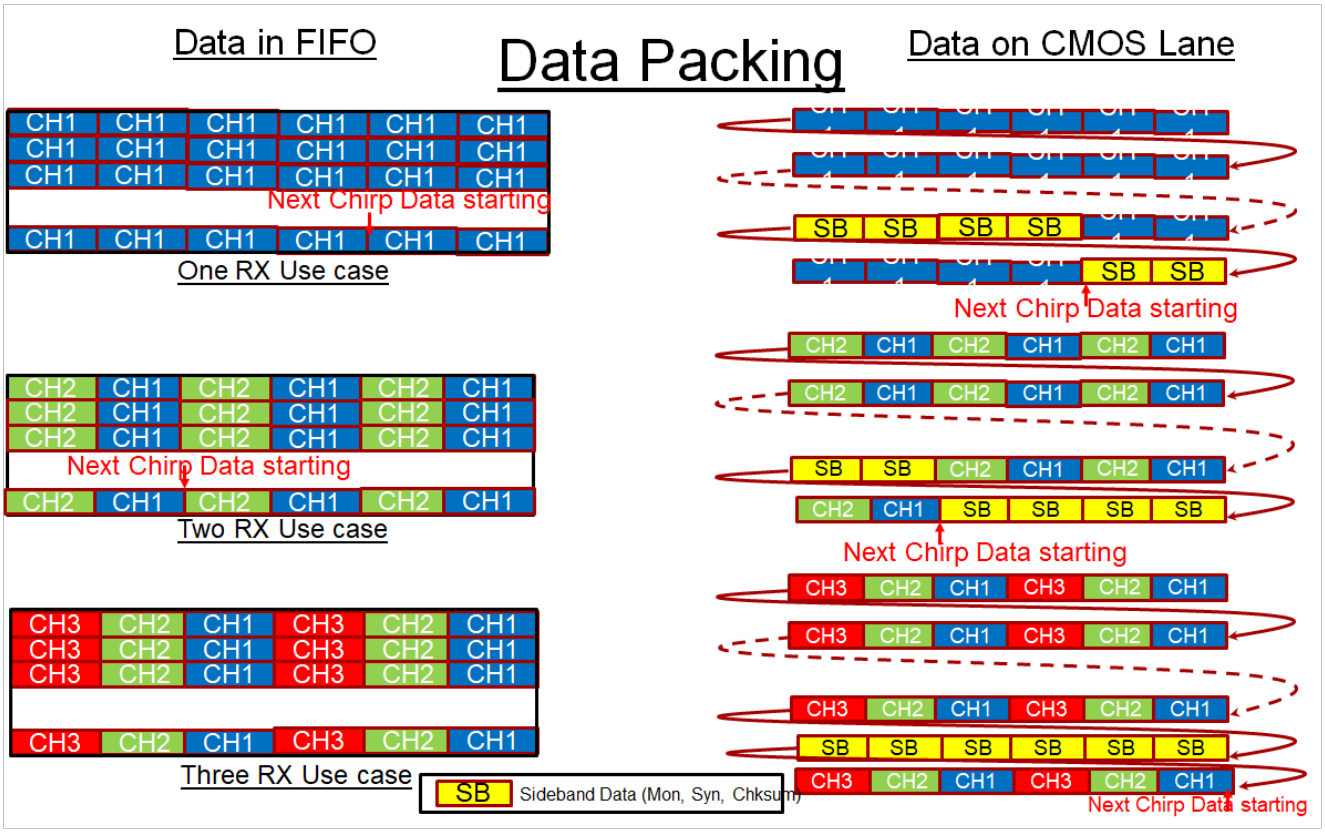


Figure 20-6. RDIF Output Lane Format

20.9 Configuration

RDIF can be configured using DFP APIs only.

20.9.1 Data Swizzling

	"00"			"01"			"10"			"11"		
	cyc#1	cyc#2	cyc#3	cyc#1	cyc#2	cyc#3	cyc#1	cyc#2	cyc#3	cyc#1	cyc#2	cyc#3
pin3	3	7	11	0	4	8	11	7	3	8	4	0
pin2	2	6	10	1	5	9	10	6	2	9	5	1
pin1	1	5	9	2	6	10	9	5	1	10	6	2
pin0	0	4	8	3	7	11	8	4	0	11	7	3

Figure 20-7. Configuration for Data Swizzling

This option applies to all the channel data and the side band data.

- It is not possible to swizzle only chirp data or side band data
- It is not possible to swizzle chirp data and side band data differently.
- Within side band data, it is not possible to swizzle each of the count or crc separately. They all have the use the same configuration.

21 On Chip Debug

21.1 DebugSS Architecture

21.1.1 Overview

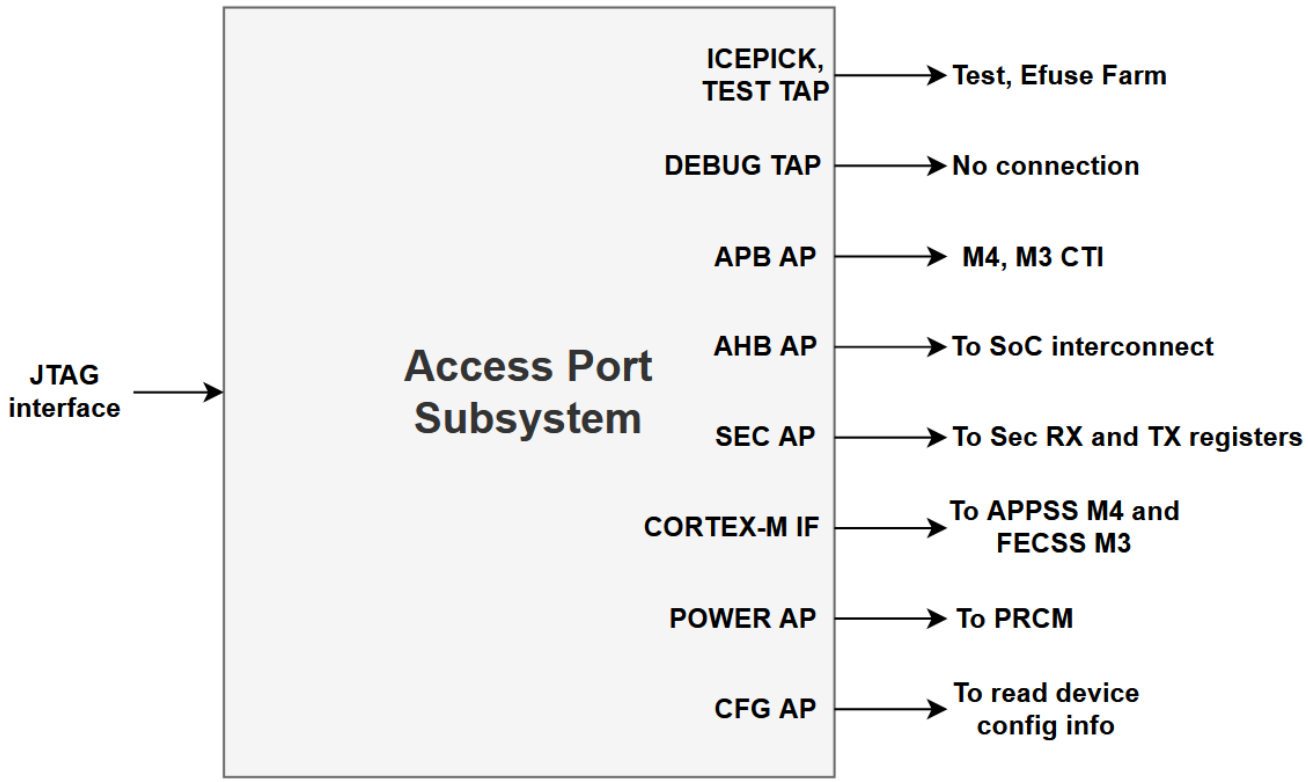


Figure 21-1. Access Port Subsystem

The OneMCU Debug Subsystem (OneMCUDebugSS) is used in the xWRLx432 platform. An overview of the interconnectivity of the debug ports is depicted in Figure 21-2.

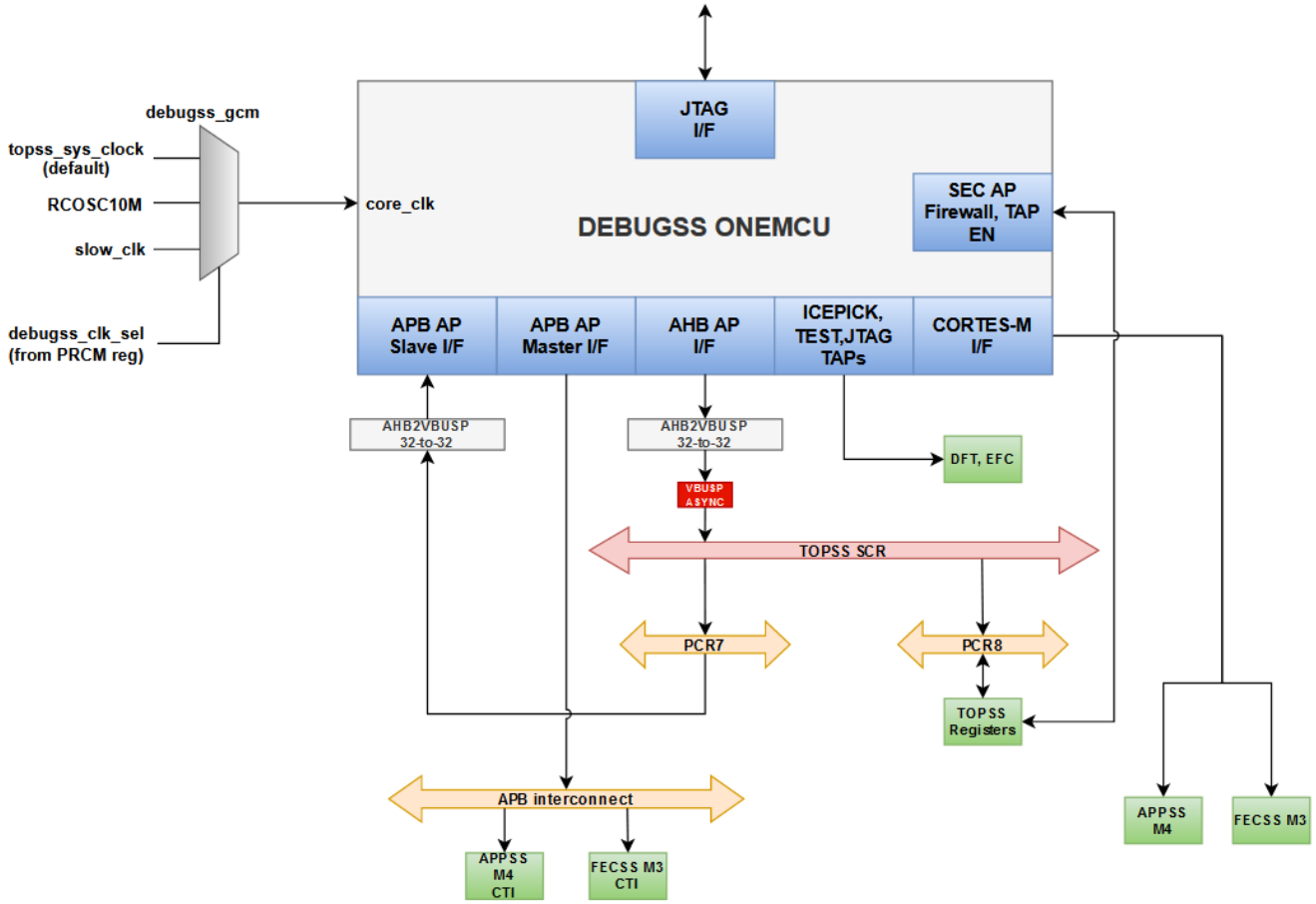


Figure 21-2. Onemcu DebugSS Integration

ADVANCE INFORMATION

21.1.2 Cross Trigger

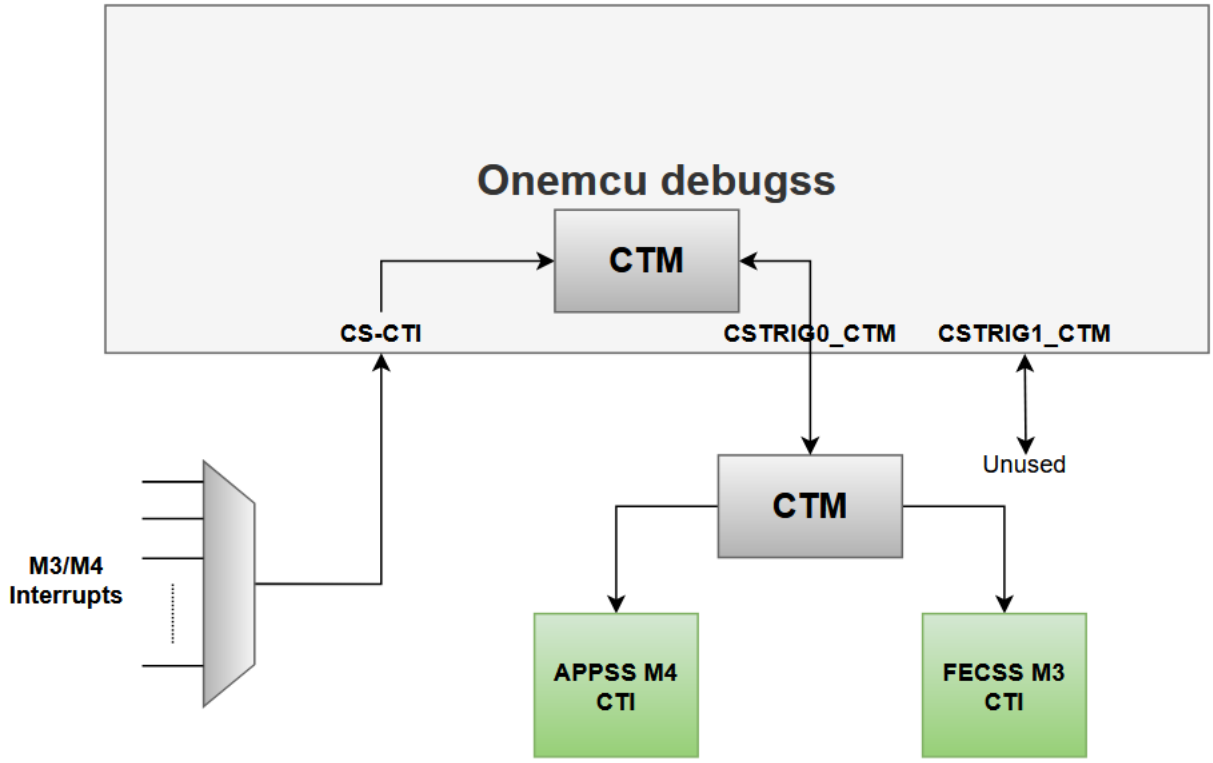


Figure 21-3. Cross Trigger Integration

21.1.3 PERIPHERAL SUSPEND CONNECTIONS

The debug extensions enable you to force the core to be stopped and placed in debug state by:

- a given instruction fetch (breakpoint)
- a data access (watchpoint)
- an external debug request

In the debug state, the core (Cortex-M3/M4) and processor memory system are effectively stopped and isolated from the rest of the system. This is known as ***halt mode*** operation and enables you to examine the internal state of the core and external AHB state, while all other system activity continues as normal. When debug has been completed, the core restores the core and system state, and resumes program execution. Some of the peripherals should not continue to operate while the core is in ***halt mode***. For example, Watch Dog Timer (WDT) should ***suspend*** some of its operations when the core is in ***halt mode***, else the Watch Dog Timer would timeout while the core is not running. The following peripherals support ***halt mode***, and disable some of their operation when the core is in ***halt mode***. Refer to individual peripheral chapters in the user manual for ***suspend*** features of the respective peripherals.

- APP_SS :: RTI
- APP_SS :: WDT
- MCRC
- I2C
- SCI/UART
- MCAN
- LIN
- EPWM

When the core is in ***halt mode***, the examination of the internal state of the core uses a JTAG-style interface, that enables you to serially insert instructions into the instruction pipeline. This exports the contents of the core registers. The exported data is serially shifted out without affecting the rest of the system.

When the core is in halt mode, the Cortex-M3/M4 has an output signal named "***HALTED***" and is used to connect to the peripherals that supports.

21.1.4 Address Map

Table 21-1. Address Map for DebugSS

APB Port	Block Name	Start Address Offset	End Address Offset
APB EXTERNAL PORT 0	APPSS CM4 CTI	0x00010000	0x00010FFF
APB EXTERNAL PORT 0	FECSS CM4 CTI	0x00011000	0x00011FFF

21.1.5 Debug Bus Integration

TOP Level Integration:

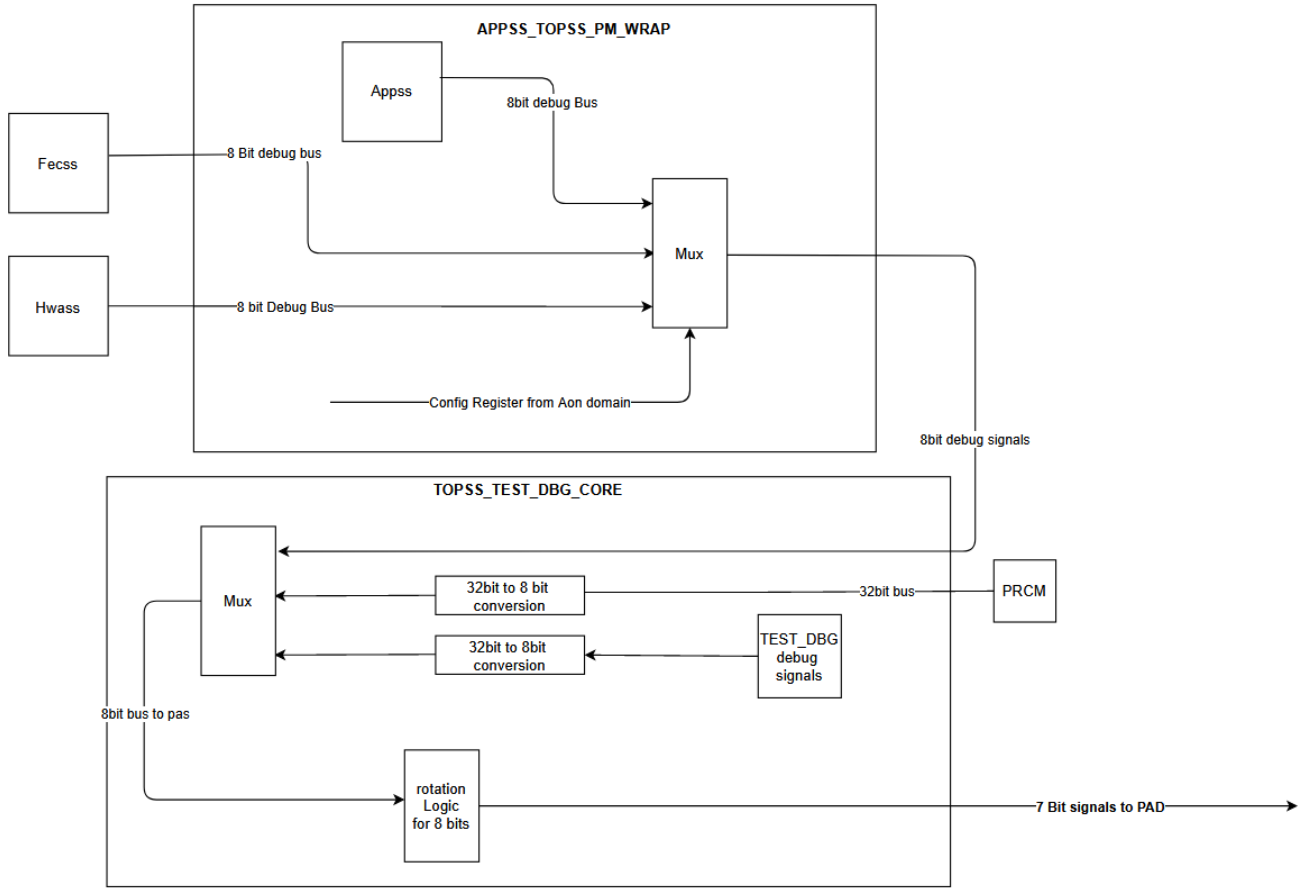


Figure 21-4. Debug Bus

ADVANCE INFORMATION

21.2 TOP_DEBUGSS Registers

Table 21-2 lists the memory-mapped registers for the TOP_DEBUGSS registers. All register offset addresses not listed in Table 21-2 should be considered as reserved locations and the register contents should not be modified.

Table 21-2. TOP_DEBUGSS Registers

Offset	Acronym	Register Name	Section
0h	ONEMCU_APB_BASE	Start Address of ROM Table	Go
FFCh	ONEMCU_APB_BASE_END	End Address of ROM Table	Go
1000h	ONEMCU_CTI_CONTROL	http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdjefbi.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdefejc.html	Go
1010h	ONEMCU_CTI_INTACK		Go
1014h	ONEMCU_CTI_APPSET		Go
1018h	ONEMCU_CTI_APPCLEAR		Go
101Ch	ONEMCU_CTI_APPPULSE		Go
1020h	ONEMCU_CTI_INEN0		Go
1024h	ONEMCU_CTI_INEN1		Go
1028h	ONEMCU_CTI_INEN2		Go
102Ch	ONEMCU_CTI_INEN3		Go
1030h	ONEMCU_CTI_INEN4		Go
1034h	ONEMCU_CTI_INEN5		Go
1038h	ONEMCU_CTI_INEN6		Go
103Ch	ONEMCU_CTI_INEN7		Go
10A0h	ONEMCU_CTI_OUTEN0		Go
10A4h	ONEMCU_CTI_OUTEN1		Go
10A8h	ONEMCU_CTI_OUTEN2		Go
10ACh	ONEMCU_CTI_OUTEN3		Go
10B0h	ONEMCU_CTI_OUTEN4		Go
10B4h	ONEMCU_CTI_OUTEN5		Go
10B8h	ONEMCU_CTI_OUTEN6		Go
10BCh	ONEMCU_CTI_OUTEN7		Go
1130h	ONEMCU_CTI_TRIGINSTATUS		Go
1134h	ONEMCU_CTI_TRIGOUTSTATUS		Go
1138h	ONEMCU_CTI_CHINSTATUS		Go
113Ch	ONEMCU_CTI_CHOUTSTATUS		Go
1140h	ONEMCU_CTI_GATE		Go
1144h	ONEMCU_CTI_ASICCTL		Go
1EDCh	ONEMCU_CTI_ITCHINACK		Go
1EE0h	ONEMCU_CTI_ITTRIGINACK		Go
1EE4h	ONEMCU_CTI_ITCHOUT		Go
1EE8h	ONEMCU_CTI_ITTRIGOUT		Go
1EECh	ONEMCU_CTI_ITCHOUTACK		Go
1EF0h	ONEMCU_CTI_ITTRIGOUTACK		Go
1EF4h	ONEMCU_CTI_ITCHIN		Go
1EF8h	ONEMCU_CTI_ITTRIGIN		Go
1F00h	ONEMCU_CTI_ITCTRL		Go
1FA0h	ONEMCU_CTI_Claim_Tag_Set		Go
1FA4h	ONEMCU_CTI_Claim_Tag_Clear		Go
1FB0h	ONEMCU_CTI_Lock_Access_Register		Go

Table 21-2. TOP_DEBUGSS Registers (continued)

Offset	Acronym	Register Name	Section
1FB4h	ONEMCU_CTI_Lock_Status_Register		Go
1FB8h	ONEMCU_CTI_Authentication_Status		Go
1FC8h	ONEMCU_CTI_Device_ID		Go
1FCCh	ONEMCU_CTI_Device_Type_Identifier		Go
1FD0h	ONEMCU_CTI_PeripheralID4		Go
1FD4h	ONEMCU_CTI_PeripheralID5		Go
1FD8h	ONEMCU_CTI_PeripheralID6		Go
1FDCh	ONEMCU_CTI_PeripheralID7		Go
1FE0h	ONEMCU_CTI_PeripheralID0		Go
1FE4h	ONEMCU_CTI_PeripheralID1		Go
1FE8h	ONEMCU_CTI_PeripheralID2		Go
1FECh	ONEMCU_CTI_PeripheralID3		Go
1FF0h	ONEMCU_CTI_Component_ID0		Go
1FF4h	ONEMCU_CTI_Component_ID1		Go
1FF8h	ONEMCU_CTI_Component_ID2		Go
1FFCh	ONEMCU_CTI_Component_ID3		Go
2000h	ONEMCU_TPIU_SPORTSZ	Supported port sizes	Go
2004h	ONEMCU_TPIU_CPORTSZ	Current port size	Go
2100h	ONEMCU_TPIU_STRIGM	Supported trigger modes	Go
2104h	ONEMCU_TPIU_TRIGCNT	Trigger counter value	Go
2108h	ONEMCU_TPIU_TRIGMUL	Trigger multiplier	Go
2200h	ONEMCU_TPIU_STSTPTRN	Supported test pattern/modes	Go
2204h	ONEMCU_TPIU_CTSTPTRN	Current test pattern/mode	Go
2208h	ONEMCU_TPIU_TPRCNTR	Test pattern repeat counter	Go
2300h	ONEMCU_TPIU_FFSTS	Formatter and flush status	Go
2304h	ONEMCU_TPIU_FFCTRL	Formatter and flush control	Go
2308h	ONEMCU_TPIU_FSCNTR	Formatter synchronization counter	Go
2400h	ONEMCU_TPIU_EXCTLIN	EXTCTL In Port	Go
2404h	ONEMCU_TPIU_EXCTLOUT	EXTCTL Out Port	Go
2EE4h	ONEMCU_TPIU_ITTRFLINACK	Integration Register, ITTRFLINACK	Go
2EE8h	ONEMCU_TPIU_ITTRFLIN	Integration Register, ITTRFLIN	Go
2EECh	ONEMCU_TPIU_ITATBDATA0	Integration Register, ITATBDATA0	Go
2EF0h	ONEMCU_TPIU_ITATBCTR2	Integration Register, ITATBCTR2	Go
2EF4h	ONEMCU_TPIU_ITATBCTR1	Integration Register, ITATBCTR1	Go
2EF8h	ONEMCU_TPIU_ITATBCTR0	Integration Register, ITATBCTR0	Go
2F00h	ONEMCU_TPIU_ITCTRL	Integration Mode Control Register	Go
2FA0h	ONEMCU_TPIU_CLAIMSET	Claim Tag Set	Go
2FA4h	ONEMCU_TPIU_CLAIMCLR	Claim Tag Clear	Go
2FB0h	ONEMCU_TPIU_LAR	Lock status	Go
2FB4h	ONEMCU_TPIU_LSR	Lock Access	Go
2FB8h	ONEMCU_TPIU_AUTHSTATUS	Authentication status	Go
2FC8h	ONEMCU_TPIU_DEVID	Device ID	Go
2FCCh	ONEMCU_TPIU_DEVTYPE	Device type identifier	Go
2FD0h	ONEMCU_TPIU_PIDR4	Peripheral ID4	Go
2FD4h	ONEMCU_TPIU_PIDR5	Peripheral ID5	Go
2FD8h	ONEMCU_TPIU_PIDR6	Peripheral ID6	Go
2FDCh	ONEMCU_TPIU_PIDR7	Peripheral ID7	Go

Table 21-2. TOP_DEBUGSS Registers (continued)

Offset	Acronym	Register Name	Section
2FE0h	ONEMCU_TPIU_PIDR0	Peripheral ID0	Go
2FE4h	ONEMCU_TPIU_PIDR1	Peripheral ID1	Go
2FE8h	ONEMCU_TPIU_PIDR2	Peripheral ID2	Go
2FECh	ONEMCU_TPIU_PIDR3	Peripheral ID3	Go
2FF0h	ONEMCU_TPIU_CIDR0	Component ID0	Go
2FF4h	ONEMCU_TPIU_CIDR1	Component ID1	Go
2FF8h	ONEMCU_TPIU_CIDR2	Component ID2	Go
2FFCh	ONEMCU_TPIU_CIDR3	Component ID3	Go
00010000h	APP_CM4_CTI_CONTROL	http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdjefbi.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdefejc.html	Go
00010010h	APP_CM4_CTI_INTACK		Go
00010014h	APP_CM4_CTI_APPSET		Go
00010018h	APP_CM4_CTI_APPCLEAR		Go
0001001Ch	APP_CM4_CTI_APPPULSE		Go
00010020h	APP_CM4_CTI_INEN0		Go
00010024h	APP_CM4_CTI_INEN1		Go
00010028h	APP_CM4_CTI_INEN2		Go
0001002Ch	APP_CM4_CTI_INEN3		Go
00010030h	APP_CM4_CTI_INEN4		Go
00010034h	APP_CM4_CTI_INEN5		Go
00010038h	APP_CM4_CTI_INEN6		Go
0001003Ch	APP_CM4_CTI_INEN7		Go
000100A0h	APP_CM4_CTI_OUTEN0		Go
000100A4h	APP_CM4_CTI_OUTEN1		Go
000100A8h	APP_CM4_CTI_OUTEN2		Go
000100ACh	APP_CM4_CTI_OUTEN3		Go
000100B0h	APP_CM4_CTI_OUTEN4		Go
000100B4h	APP_CM4_CTI_OUTEN5		Go
000100B8h	APP_CM4_CTI_OUTEN6		Go
000100BCh	APP_CM4_CTI_OUTEN7		Go
00010130h	APP_CM4_CTI_TRIGINSTATUS		Go
00010134h	APP_CM4_CTI_TRIGOUTSTATUS		Go
00010138h	APP_CM4_CTI_CHINSTATUS		Go
0001013Ch	APP_CM4_CTI_CHOUTSTATUS		Go
00010140h	APP_CM4_CTI_GATE		Go
00010144h	APP_CM4_CTI_ASICCTL		Go
00010EDCh	APP_CM4_CTI_ITCHINACK		Go
00010EE0h	APP_CM4_CTI_ITTRIGINACK		Go
00010EE4h	APP_CM4_CTI_ITCHOUT		Go
00010EE8h	APP_CM4_CTI_ITTRIGOUT		Go
00010EECh	APP_CM4_CTI_ITCHOUTACK		Go
00010EF0h	APP_CM4_CTI_ITTRIGOUTACK		Go
00010EF4h	APP_CM4_CTI_ITCHIN		Go
00010EF8h	APP_CM4_CTI_ITTRIGIN		Go
00010F00h	APP_CM4_CTI_ITCTRL		Go
00010FA0h	APP_CM4_CTI_Claim_Tag_Set		Go

Table 21-2. TOP_DEBUGSS Registers (continued)

Offset	Acronym	Register Name	Section
00010FA4h	APP_CM4_CTI_Claim_Tag_Clear		Go
00010FB0h	APP_CM4_CTI_Lock_Access_Register		Go
00010FB4h	APP_CM4_CTI_Lock_Status_Register		Go
00010FB8h	APP_CM4_CTI_Authentication_Status		Go
00010FC8h	APP_CM4_CTI_Device_ID		Go
00010FCCh	APP_CM4_CTI_Device_Type_Identifier		Go
00010FD0h	APP_CM4_CTI_PeripheralID4		Go
00010FD4h	APP_CM4_CTI_PeripheralID5		Go
00010FD8h	APP_CM4_CTI_PeripheralID6		Go
00010FDCh	APP_CM4_CTI_PeripheralID7		Go
00010FE0h	APP_CM4_CTI_PeripheralID0		Go
00010FE4h	APP_CM4_CTI_PeripheralID1		Go
00010FE8h	APP_CM4_CTI_PeripheralID2		Go
00010FECh	APP_CM4_CTI_PeripheralID3		Go
00010FF0h	APP_CM4_CTI_Component_ID0		Go
00010FF4h	APP_CM4_CTI_Component_ID1		Go
00010FF8h	APP_CM4_CTI_Component_ID2		Go
00010FFCh	APP_CM4_CTI_Component_ID3		Go
00011000h	FEC_CM3_CTI_CONTROL	http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdjefbi.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdefejc.html	Go
00011010h	FEC_CM3_CTI_INTACK		Go
00011014h	FEC_CM3_CTI_APPSET		Go
00011018h	FEC_CM3_CTI_APPCLEAR		Go
0001101Ch	FEC_CM3_CTI_APPPULSE		Go
00011020h	FEC_CM3_CTI_INEN0		Go
00011024h	FEC_CM3_CTI_INEN1		Go
00011028h	FEC_CM3_CTI_INEN2		Go
0001102Ch	FEC_CM3_CTI_INEN3		Go
00011030h	FEC_CM3_CTI_INEN4		Go
00011034h	FEC_CM3_CTI_INEN5		Go
00011038h	FEC_CM3_CTI_INEN6		Go
0001103Ch	FEC_CM3_CTI_INEN7		Go
000110A0h	FEC_CM3_CTI_OUTEN0		Go
000110A4h	FEC_CM3_CTI_OUTEN1		Go
000110A8h	FEC_CM3_CTI_OUTEN2		Go
000110ACh	FEC_CM3_CTI_OUTEN3		Go
000110B0h	FEC_CM3_CTI_OUTEN4		Go
000110B4h	FEC_CM3_CTI_OUTEN5		Go
000110B8h	FEC_CM3_CTI_OUTEN6		Go
000110BCh	FEC_CM3_CTI_OUTEN7		Go
00011130h	FEC_CM3_CTI_TRIGINSTATUS		Go
00011134h	FEC_CM3_CTI_TRIGOUTSTATUS		Go
00011138h	FEC_CM3_CTI_CHINSTATUS		Go
0001113Ch	FEC_CM3_CTI_CHOUTSTATUS		Go
00011140h	FEC_CM3_CTI_GATE		Go
00011144h	FEC_CM3_CTI_ASICCTL		Go

Table 21-2. TOP_DEBUGSS Registers (continued)

Offset	Acronym	Register Name	Section
00011EDCh	FEC_CM3_CTI_ITCHINACK		Go
00011EE0h	FEC_CM3_CTI_ITTRIGINACK		Go
00011EE4h	FEC_CM3_CTI_ITCHOUT		Go
00011EE8h	FEC_CM3_CTI_ITTRIGOUT		Go
00011EECh	FEC_CM3_CTI_ITCHOUTACK		Go
00011EF0h	FEC_CM3_CTI_ITTRIGOUTACK		Go
00011EF4h	FEC_CM3_CTI_ITCHIN		Go
00011EF8h	FEC_CM3_CTI_ITTRIGIN		Go
00011F00h	FEC_CM3_CTI_ITCTRL		Go
00011FA0h	FEC_CM3_CTI_Claim_Tag_Set		Go
00011FA4h	FEC_CM3_CTI_Claim_Tag_Clear		Go
00011FB0h	FEC_CM3_CTI_Lock_Access_Register		Go
00011FB4h	FEC_CM3_CTI_Lock_Status_Register		Go
00011FB8h	FEC_CM3_CTI_Authentication_Status		Go
00011FC8h	FEC_CM3_CTI_Device_ID		Go
00011FCCh	FEC_CM3_CTI_Device_Type_Identifier		Go
00011FD0h	FEC_CM3_CTI_PeripheralID4		Go
00011FD4h	FEC_CM3_CTI_PeripheralID5		Go
00011FD8h	FEC_CM3_CTI_PeripheralID6		Go
00011FDCh	FEC_CM3_CTI_PeripheralID7		Go
00011FE0h	FEC_CM3_CTI_PeripheralID0		Go
00011FE4h	FEC_CM3_CTI_PeripheralID1		Go
00011FE8h	FEC_CM3_CTI_PeripheralID2		Go
00011FECh	FEC_CM3_CTI_PeripheralID3		Go
00011FF0h	FEC_CM3_CTI_Component_ID0		Go
00011FF4h	FEC_CM3_CTI_Component_ID1		Go
00011FF8h	FEC_CM3_CTI_Component_ID2		Go
00011FFCh	FEC_CM3_CTI_Component_ID3		Go

Complex bit access types are encoded to fit into small table cells. Table 21-3 shows the codes that are used for access types in this section.

Table 21-3. TOP_DEBUGSS 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

21.2.1 ONEMCU_APB_BASE Register (Offset = 0h) [Reset = 00000000h]

ONEMCU_APB_BASE is shown in Table 21-4.

Return to the Summary Table.

Start Address of ROM Table

Table 21-4. ONEMCU_APB_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_APB_BASE	R	0h	OneMCU APB Space : Start Address of ROM Table

21.2.2 ONEMCU_APB_BASE_END Register (Offset = FFCh) [Reset = 00000000h]

ONEMCU_APB_BASE_END is shown in Table 21-5.

Return to the Summary Table.

End Address of ROM Table

Table 21-5. ONEMCU_APB_BASE_END Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_APB_BASE_END	R	0h	OneMCU APB Space : Endt Address of ROM Table

21.2.3 ONEMCU_CTI_CONTROL Register (Offset = 1000h) [Reset = 00000000h]

ONEMCU_CTI_CONTROL is shown in Table 21-6.

Return to the Summary Table.

http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdjefbi.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdefejc.html

Table 21-6. ONEMCU_CTI_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_CONTROL	R/W	0h	http://infocenter.arm.com/help/topic/com.arm.doc.ddi0480e/CHDGDIHE.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0480e/CHDHBDIA.html

21.2.4 ONEMCU_CTI_INTACK Register (Offset = 1010h) [Reset = 00000000h]

ONEMCU_CTI_INTACK is shown in Table 21-7.

Return to the Summary Table.

Table 21-7. ONEMCU_CTI_INTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INTACK	W	0h	

21.2.5 ONEMCU_CTI_APPSET Register (Offset = 1014h) [Reset = 00000000h]

ONEMCU_CTI_APPSET is shown in Table 21-8.

Return to the Summary Table.

Table 21-8. ONEMCU_CTI_APPSET Register Field Descriptions

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

21.2.6 ONEMCU_CTI_APPCLEAR Register (Offset = 1018h) [Reset = 00000000h]

ONEMCU_CTI_APPCLEAR is shown in Table 21-9.

Return to the Summary Table.

Table 21-9. ONEMCU_CTI_APPCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_APPCLEAR	W	0h	

21.2.7 ONEMCU_CTI_APPPULSE Register (Offset = 101Ch) [Reset = 00000000h]

ONEMCU_CTI_APPPULSE is shown in Table 21-10.

Return to the Summary Table.

Table 21-10. ONEMCU_CTI_APPPULSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_APPPULSE	W	0h	

21.2.8 ONEMCU_CTI_INEN0 Register (Offset = 1020h) [Reset = 00000000h]

ONEMCU_CTI_INEN0 is shown in Table 21-11.

Return to the Summary Table.

Table 21-11. ONEMCU_CTI_INEN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN0	R/W	0h	

21.2.9 ONEMCU_CTI_INEN1 Register (Offset = 1024h) [Reset = 00000000h]

ONEMCU_CTI_INEN1 is shown in Table 21-12.

Return to the Summary Table.

Table 21-12. ONEMCU_CTI_INEN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN1	R/W	0h	

21.2.10 ONEMCU_CTI_INEN2 Register (Offset = 1028h) [Reset = 00000000h]

ONEMCU_CTI_INEN2 is shown in Table 21-13.

Return to the Summary Table.

Table 21-13. ONEMCU_CTI_INEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN2	R/W	0h	

21.2.11 ONEMCU_CTI_INEN3 Register (Offset = 102Ch) [Reset = 00000000h]

ONEMCU_CTI_INEN3 is shown in Table 21-14.

Return to the Summary Table.

Table 21-14. ONEMCU_CTI_INEN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN3	R/W	0h	

21.2.12 ONEMCU_CTI_INEN4 Register (Offset = 1030h) [Reset = 00000000h]

ONEMCU_CTI_INEN4 is shown in Table 21-15.

Return to the Summary Table.

Table 21-15. ONEMCU_CTI_INEN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN4	R/W	0h	

21.2.13 ONEMCU_CTI_INEN5 Register (Offset = 1034h) [Reset = 00000000h]

ONEMCU_CTI_INEN5 is shown in Table 21-16.

Return to the Summary Table.

Table 21-16. ONEMCU_CTI_INEN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN5	R/W	0h	

21.2.14 ONEMCU_CTI_INEN6 Register (Offset = 1038h) [Reset = 00000000h]

ONEMCU_CTI_INEN6 is shown in Table 21-17.

Return to the Summary Table.

Table 21-17. ONEMCU_CTI_INEN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN6	R/W	0h	

21.2.15 ONEMCU_CTI_INEN7 Register (Offset = 103Ch) [Reset = 00000000h]

ONEMCU_CTI_INEN7 is shown in Table 21-18.

Return to the Summary Table.

Table 21-18. ONEMCU_CTI_INEN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_INEN7	R/W	0h	

21.2.16 ONEMCU_CTI_OUTEN0 Register (Offset = 10A0h) [Reset = 00000000h]

ONEMCU_CTI_OUTEN0 is shown in Table 21-19.

Return to the Summary Table.

Table 21-19. ONEMCU_CTI_OUTEN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN0	R/W	0h	

21.2.17 ONEMCU_CTI_OUTEN1 Register (Offset = 10A4h) [Reset = 00000000h]

ONEMCU_CTI_OUTEN1 is shown in Table 21-20.

Return to the Summary Table.

Table 21-20. ONEMCU_CTI_OUTEN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN1	R/W	0h	

21.2.18 ONEMCU_CTI_OUTEN2 Register (Offset = 10A8h) [Reset = 00000000h]

ONEMCU_CTI_OUTEN2 is shown in Table 21-21.

Return to the Summary Table.

Table 21-21. ONEMCU_CTI_OUTEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN2	R/W	0h	

21.2.19 ONEMCU_CTI_OUTEN3 Register (Offset = 10ACh) [Reset = 00000000h]

ONEMCU_CTI_OUTEN3 is shown in Table 21-22.

Return to the Summary Table.

Table 21-22. ONEMCU_CTI_OUTEN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN3	R/W	0h	

21.2.20 ONEMCU_CTI_OUTEN4 Register (Offset = 10B0h) [Reset = 00000000h]

ONEMCU_CTI_OUTEN4 is shown in Table 21-23.

Return to the Summary Table.

Table 21-23. ONEMCU_CTI_OUTEN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN4	R/W	0h	

21.2.21 ONEMCU_CTI_OUTEN5 Register (Offset = 10B4h) [Reset = 00000000h]

ONEMCU_CTI_OUTEN5 is shown in Table 21-24.

Return to the Summary Table.

Table 21-24. ONEMCU_CTI_OUTEN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN5	R/W	0h	

21.2.22 ONEMCU_CTI_OUTEN6 Register (Offset = 10B8h) [Reset = 00000000h]

ONEMCU_CTI_OUTEN6 is shown in Table 21-25.

Return to the Summary Table.

Table 21-25. ONEMCU_CTI_OUTEN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN6	R/W	0h	

21.2.23 ONEMCU_CTI_OUTEN7 Register (Offset = 10BCh) [Reset = 00000000h]

ONEMCU_CTI_OUTEN7 is shown in Table 21-26.

Return to the Summary Table.

Table 21-26. ONEMCU_CTI_OUTEN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_OUTEN7	R/W	0h	

21.2.24 ONEMCU_CTI_TRIGINSTATUS Register (Offset = 1130h) [Reset = 00000000h]

ONEMCU_CTI_TRIGINSTATUS is shown in Table 21-27.

Return to the Summary Table.

Table 21-27. ONEMCU_CTI_TRIGINSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_TRIGINSTATUS	R	0h	

21.2.25 ONEMCU_CTI_TRIGOUTSTATUS Register (Offset = 1134h) [Reset = 00000000h]

ONEMCU_CTI_TRIGOUTSTATUS is shown in Table 21-28.

Return to the Summary Table.

Table 21-28. ONEMCU_CTI_TRIGOUTSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_TRIGOUTSTATUS	R	0h	

21.2.26 ONEMCU_CTI_CHINSTATUS Register (Offset = 1138h) [Reset = 00000000h]

ONEMCU_CTI_CHINSTATUS is shown in Table 21-29.

Return to the Summary Table.

Table 21-29. ONEMCU_CTI_CHINSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_CHINSTATUS	R	0h	

21.2.27 ONEMCU_CTI_CHOUTSTATUS Register (Offset = 113Ch) [Reset = 00000000h]

ONEMCU_CTI_CHOUTSTATUS is shown in Table 21-30.

Return to the Summary Table.

Table 21-30. ONEMCU_CTI_CHOUTSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_CHOUTSTATUS	R	0h	

21.2.28 ONEMCU_CTI_GATE Register (Offset = 1140h) [Reset = 00000000h]

ONEMCU_CTI_GATE is shown in Table 21-31.

Return to the Summary Table.

Table 21-31. ONEMCU_CTI_GATE Register Field Descriptions

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

21.2.29 ONEMCU_CTI_ASICCTL Register (Offset = 1144h) [Reset = 00000000h]

ONEMCU_CTI_ASICCTL is shown in Table 21-32.

Return to the Summary Table.

Table 21-32. ONEMCU_CTI_ASICCTL Register Field Descriptions

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

21.2.30 ONEMCU_CTI_ITCHINACK Register (Offset = 1EDCh) [Reset = 00000000h]

ONEMCU_CTI_ITCHINACK is shown in Table 21-33.

Return to the Summary Table.

Table 21-33. ONEMCU_CTI_ITCHINACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITCHINACK	W	0h	

21.2.31 ONEMCU_CTI_ITTRIGINACK Register (Offset = 1EE0h) [Reset = 00000000h]

ONEMCU_CTI_ITTRIGINACK is shown in Table 21-34.

Return to the Summary Table.

Table 21-34. ONEMCU_CTI_ITTRIGINACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITTRIGINACK	W	0h	

21.2.32 ONEMCU_CTI_ITCHOUT Register (Offset = 1EE4h) [Reset = 00000000h]

ONEMCU_CTI_ITCHOUT is shown in Table 21-35.

Return to the Summary Table.

Table 21-35. ONEMCU_CTI_ITCHOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITCHOUT	W	0h	

21.2.33 ONEMCU_CTI_ITTRIGOUT Register (Offset = 1EE8h) [Reset = 00000000h]

ONEMCU_CTI_ITTRIGOUT is shown in Table 21-36.

Return to the Summary Table.

Table 21-36. ONEMCU_CTI_ITTRIGOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITTRIGOUT	W	0h	

21.2.34 ONEMCU_CTI_ITCHOUTACK Register (Offset = 1EECh) [Reset = 00000000h]

ONEMCU_CTI_ITCHOUTACK is shown in Table 21-37.

Return to the Summary Table.

Table 21-37. ONEMCU_CTI_ITCHOUTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITCHOUTACK	R	0h	

21.2.35 ONEMCU_CTI_ITTRIGOUTACK Register (Offset = 1EF0h) [Reset = 00000000h]

ONEMCU_CTI_ITTRIGOUTACK is shown in Table 21-38.

Return to the Summary Table.

Table 21-38. ONEMCU_CTI_ITTRIGOUTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITTRIGOUTACK	R	0h	

21.2.36 ONEMCU_CTI_ITCHIN Register (Offset = 1EF4h) [Reset = 00000000h]

ONEMCU_CTI_ITCHIN is shown in Table 21-39.

Return to the Summary Table.

Table 21-39. ONEMCU_CTI_ITCHIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITCHIN	R	0h	

21.2.37 ONEMCU_CTI_ITTRIGIN Register (Offset = 1EF8h) [Reset = 00000000h]

ONEMCU_CTI_ITTRIGIN is shown in Table 21-40.

Return to the Summary Table.

Table 21-40. ONEMCU_CTI_ITTRIGIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_ITTRIGIN	R	0h	

21.2.38 ONEMCU_CTI_ITCTRL Register (Offset = 1F00h) [Reset = 00000000h]

ONEMCU_CTI_ITCTRL is shown in Table 21-41.

Return to the Summary Table.

Table 21-41. ONEMCU_CTI_ITCTRL Register Field Descriptions

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

21.2.39 ONEMCU_CTI_Claim_Tag_Set Register (Offset = 1FA0h) [Reset = 00000000h]

ONEMCU_CTI_Claim_Tag_Set is shown in Table 21-42.

Return to the Summary Table.

Table 21-42. ONEMCU_CTI_Claim_Tag_Set Register Field Descriptions

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

21.2.40 ONEMCU_CTI_Claim_Tag_Clear Register (Offset = 1FA4h) [Reset = 00000000h]

ONEMCU_CTI_Claim_Tag_Clear is shown in Table 21-43.

Return to the Summary Table.

Table 21-43. ONEMCU_CTI_Claim_Tag_Clear Register Field Descriptions

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

21.2.41 ONEMCU_CTI_Lock_Access_Register (Offset = 1FB0h) [Reset = 00000000h]

ONEMCU_CTI_Lock_Access_Register is shown in Table 21-44.

Return to the Summary Table.

Table 21-44. ONEMCU_CTI_Lock_Access_Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Lock_Access_Register	W	0h	

21.2.42 ONEMCU_CTI_Lock_Status_Register (Offset = 1FB4h) [Reset = 00000000h]

ONEMCU_CTI_Lock_Status_Register is shown in Table 21-45.

Return to the Summary Table.

Table 21-45. ONEMCU_CTI_Lock_Status_Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Lock_Status_Register	R	0h	

21.2.43 ONEMCU_CTI_Authentication_Status Register (Offset = 1FB8h) [Reset = 00000000h]

ONEMCU_CTI_Authentication_Status is shown in Table 21-46.

Return to the Summary Table.

Table 21-46. ONEMCU_CTI_Authentication_Status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Authentication_Status	R	0h	

21.2.44 ONEMCU_CTI_Device_ID Register (Offset = 1FC8h) [Reset = 00000000h]

ONEMCU_CTI_Device_ID is shown in Table 21-47.

Return to the Summary Table.

Table 21-47. ONEMCU_CTI_Device_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Device_ID	R	0h	

21.2.45 ONEMCU_CTI_Device_Type_Identifier Register (Offset = 1FCCh) [Reset = 00000000h]

ONEMCU_CTI_Device_Type_Identifier is shown in Table 21-48.

Return to the Summary Table.

Table 21-48. ONEMCU_CTI_Device_Type_Identifier Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Device_Type_Identifier	R	0h	

21.2.46 ONEMCU_CTI_PeripheralID4 Register (Offset = 1FD0h) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID4 is shown in Table 21-49.

Return to the Summary Table.

Table 21-49. ONEMCU_CTI_PeripheralID4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_PeripheralID4	R	0h	

21.2.47 ONEMCU_CTI_PeripheralID5 Register (Offset = 1FD4h) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID5 is shown in Table 21-50.

Return to the Summary Table.

Table 21-50. ONEMCU_CTI_PeripheralID5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_PeripheralID5	R	0h	

21.2.48 ONEMCU_CTI_PeripheralID6 Register (Offset = 1FD8h) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID6 is shown in Table 21-51.

Return to the Summary Table.

Table 21-51. ONEMCU_CTI_PeripheralID6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_PeripheralID6	R	0h	

21.2.49 ONEMCU_CTI_PeripheralID7 Register (Offset = 1FDCh) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID7 is shown in Table 21-52.

Return to the Summary Table.

Table 21-52. ONEMCU_CTI_PeripheralID7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Peripheral ID7	R	0h	

21.2.50 ONEMCU_CTI_PeripheralID0 Register (Offset = 1FE0h) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID0 is shown in Table 21-53.

Return to the Summary Table.

Table 21-53. ONEMCU_CTI_PeripheralID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Peripheral ID0	R	0h	

21.2.51 ONEMCU_CTI_PeripheralID1 Register (Offset = 1FE4h) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID1 is shown in Table 21-54.

Return to the Summary Table.

Table 21-54. ONEMCU_CTI_PeripheralID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Peripheral ID1	R	0h	

21.2.52 ONEMCU_CTI_PeripheralID2 Register (Offset = 1FE8h) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID2 is shown in Table 21-55.

Return to the Summary Table.

Table 21-55. ONEMCU_CTI_PeripheralID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Peripheral ID2	R	0h	

21.2.53 ONEMCU_CTI_PeripheralID3 Register (Offset = 1FECh) [Reset = 00000000h]

ONEMCU_CTI_PeripheralID3 is shown in Table 21-56.

Return to the Summary Table.

Table 21-56. ONEMCU_CTI_PeripheralID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Peripheral ID3	R	0h	

21.2.54 ONEMCU_CTI_Component_ID0 Register (Offset = 1FF0h) [Reset = 00000000h]

ONEMCU_CTI_Component_ID0 is shown in Table 21-57.

Return to the Summary Table.

Table 21-57. ONEMCU_CTI_Component_ID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Component_ID0	R	0h	

21.2.55 ONEMCU_CTI_Component_ID1 Register (Offset = 1FF4h) [Reset = 00000000h]

ONEMCU_CTI_Component_ID1 is shown in Table 21-58.

Return to the Summary Table.

Table 21-58. ONEMCU_CTI_Component_ID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Component_ID1	R	0h	

21.2.56 ONEMCU_CTI_Component_ID2 Register (Offset = 1FF8h) [Reset = 00000000h]

ONEMCU_CTI_Component_ID2 is shown in Table 21-59.

Return to the Summary Table.

Table 21-59. ONEMCU_CTI_Component_ID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Component_ID2	R	0h	

21.2.57 ONEMCU_CTI_Component_ID3 Register (Offset = 1FFCh) [Reset = 00000000h]

ONEMCU_CTI_Component_ID3 is shown in Table 21-60.

Return to the Summary Table.

Table 21-60. ONEMCU_CTI_Component_ID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_CTI_Component_ID3	R	0h	

21.2.58 ONEMCU_TPIU_SPORTSZ Register (Offset = 2000h) [Reset = 0FFFFFFFh]

ONEMCU_TPIU_SPORTSZ is shown in Table 21-61.

Return to the Summary Table.

Supported port sizes

Table 21-61. ONEMCU_TPIU_SPORTSZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_SPORTSZ	R	0FFFFFFFh	Supported port sizes

21.2.59 ONEMCU_TPIU_CPORTSZ Register (Offset = 2004h) [Reset = 00000001h]

ONEMCU_TPIU_CPORTSZ is shown in Table 21-62.

Return to the Summary Table.

Current port size

Table 21-62. ONEMCU_TPIU_CPORTSZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CPORTSZ	R/W	00000001h	Current port size

21.2.60 ONEMCU_TPIU_STRIGM Register (Offset = 2100h) [Reset = 0000011Fh]

ONEMCU_TPIU_STRIGM is shown in Table 21-63.

Return to the Summary Table.

Supported trigger modes

Table 21-63. ONEMCU_TPIU_STRIGM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_STRIGM	R	11Fh	Supported trigger modes

21.2.61 ONEMCU_TPIU_TRIGCNT Register (Offset = 2104h) [Reset = 00000000h]

ONEMCU_TPIU_TRIGCNT is shown in Table 21-64.

Return to the Summary Table.

Trigger counter value

Table 21-64. ONEMCU_TPIU_TRIGCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_TRIGCNT	R/W	00h	Trigger counter value

21.2.62 ONEMCU_TPIU_TRIGMUL Register (Offset = 2108h) [Reset = 00000000h]

ONEMCU_TPIU_TRIGMUL is shown in Table 21-65.

Return to the Summary Table.

Trigger multiplier

Table 21-65. ONEMCU_TPIU_TRIGMUL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_TRIGMUL	R/W	00h	Trigger multiplier

21.2.63 ONEMCU_TPIU_STSTPTRN Register (Offset = 2200h) [Reset = 0003000Fh]

ONEMCU_TPIU_STSTPTRN is shown in Table 21-66.

Return to the Summary Table.

Supported test pattern/modes

Table 21-66. ONEMCU_TPIU_STSTPTRN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_STSTPTRN	R	0003000Fh	Supported test pattern/modes

21.2.64 ONEMCU_TPIU_CTSTPTRN Register (Offset = 2204h) [Reset = 00000000h]

ONEMCU_TPIU_CTSTPTRN is shown in Table 21-67.

Return to the Summary Table.

Current test pattern/mode

Table 21-67. ONEMCU_TPIU_CTSTPTRN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CTSTPTRN	R/W	00000000h	Current test pattern/mode

21.2.65 ONEMCU_TPIU_TPRCNTR Register (Offset = 2208h) [Reset = 00000000h]

ONEMCU_TPIU_TPRCNTR is shown in Table 21-68.

Return to the Summary Table.

Test pattern repeat counter

Table 21-68. ONEMCU_TPIU_TPRCNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_TPRCNTR	R/W	00h	Test pattern repeat counter

21.2.66 ONEMCU_TPIU_FFSTS Register (Offset = 2300h) [Reset = 00000002h]

ONEMCU_TPIU_FFSTS is shown in Table 21-69.

Return to the Summary Table.

Formatter and flush status

Table 21-69. ONEMCU_TPIU_FFSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_FFSTS	R	2h	Formatter and flush status

21.2.67 ONEMCU_TPIU_FFCTRL Register (Offset = 2304h) [Reset = 00001000h]

ONEMCU_TPIU_FFCTRL is shown in Table 21-70.

Return to the Summary Table.

Formatter and flush control

Table 21-70. ONEMCU_TPIU_FFCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_FFCTRL	R/W	1000h	Formatter and flush control

21.2.68 ONEMCU_TPIU_FSCNTR Register (Offset = 2308h) [Reset = 00000040h]

ONEMCU_TPIU_FSCNTR is shown in Table 21-71.

Return to the Summary Table.

Formatter synchronization counter

Table 21-71. ONEMCU_TPIU_FSCNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_FSCNTR	R/W	040h	Formatter synchronization counter

21.2.69 ONEMCU_TPIU_EXCTLIN Register (Offset = 2400h) [Reset = 00000000h]

ONEMCU_TPIU_EXCTLIN is shown in Table 21-72.

Return to the Summary Table.

EXTCTL In Port

Table 21-72. ONEMCU_TPIU_EXCTLIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_EXCTLIN	R	0h	EXTCTL In Port

21.2.70 ONEMCU_TPIU_EXCTLOUT Register (Offset = 2404h) [Reset = 00000000h]

ONEMCU_TPIU_EXCTLOUT is shown in Table 21-73.

Return to the Summary Table.

EXTCTL Out Port

Table 21-73. ONEMCU_TPIU_EXCTLOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_EXCTLOUT	R/W	00h	EXTCTL Out Port

21.2.71 ONEMCU_TPIU_ITTRFLINACK Register (Offset = 2EE4h) [Reset = 00000000h]

ONEMCU_TPIU_ITTRFLINACK is shown in Table 21-74.

Return to the Summary Table.

Integration Register, ITTRFLINACK

Table 21-74. ONEMCU_TPIU_ITTRFLINACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_ITTRFLINACK	W	0h	Integration Register, ITTRFLINACK

21.2.72 ONEMCU_TPIU_ITTRFLIN Register (Offset = 2EE8h) [Reset = 00000000h]

ONEMCU_TPIU_ITTRFLIN is shown in Table 21-75.

Return to the Summary Table.

Integration Register, ITTRFLIN

Table 21-75. ONEMCU_TPIU_ITTRFLIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_ITTRFLIN	R	0h	Integration Register, ITTRFLIN

21.2.73 ONEMCU_TPIU_ITATBDATA0 Register (Offset = 2EECh) [Reset = 00000000h]

ONEMCU_TPIU_ITATBDATA0 is shown in Table 21-76.

Return to the Summary Table.

Integration Register, ITATBDATA0

Table 21-76. ONEMCU_TPIU_ITATBDATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_ITATBDATA0	R	0h	Integration Register, ITATBDATA0

21.2.74 ONEMCU_TPIU_ITATBCTR2 Register (Offset = 2EF0h) [Reset = 00000000h]

ONEMCU_TPIU_ITATBCTR2 is shown in Table 21-77.

Return to the Summary Table.

Integration Register, ITATBCTR2

Table 21-77. ONEMCU_TPIU_ITATBCTR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_ITATBCTR2	W	0h	Integration Register, ITATBCTR2

21.2.75 ONEMCU_TPIU_ITATBCTR1 Register (Offset = 2EF4h) [Reset = 00000000h]

ONEMCU_TPIU_ITATBCTR1 is shown in Table 21-78.

Return to the Summary Table.

Integration Register, ITATBCTR1

Table 21-78. ONEMCU_TPIU_ITATBCTR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_ITATBCTR1	R	0h	Integration Register, ITATBCTR1

21.2.76 ONEMCU_TPIU_ITATBCTR0 Register (Offset = 2EF8h) [Reset = 00000000h]

ONEMCU_TPIU_ITATBCTR0 is shown in Table 21-79.

Return to the Summary Table.

Integration Register, ITATBCTR0

Table 21-79. ONEMCU_TPIU_ITATBCTR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_ITATBCTR0	R	0h	Integration Register, ITATBCTR0

21.2.77 ONEMCU_TPIU_ITCTRL Register (Offset = 2F00h) [Reset = 00000000h]

ONEMCU_TPIU_ITCTRL is shown in Table 21-80.

Return to the Summary Table.

Integration Mode Control Register

Table 21-80. ONEMCU_TPIU_ITCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_ITCTRL	R/W	0h	Integration Mode Control Register

21.2.78 ONEMCU_TPIU_CLAIMSET Register (Offset = 2FA0h) [Reset = 0000000Fh]

ONEMCU_TPIU_CLAIMSET is shown in Table 21-81.

Return to the Summary Table.

Claim Tag Set

Table 21-81. ONEMCU_TPIU_CLAIMSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CLAIMSET	R/W	Fh	Claim Tag Set

21.2.79 ONEMCU_TPIU_CLAIMCLR Register (Offset = 2FA4h) [Reset = 00000000h]

ONEMCU_TPIU_CLAIMCLR is shown in Table 21-82.

Return to the Summary Table.

Claim Tag Clear

Table 21-82. ONEMCU_TPIU_CLAIMCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CLAIMCLR	R/W	0h	Claim Tag Clear

21.2.80 ONEMCU_TPIU_LAR Register (Offset = 2FB0h) [Reset = 00000000h]

ONEMCU_TPIU_LAR is shown in Table 21-83.

Return to the Summary Table.

Lock status

Table 21-83. ONEMCU_TPIU_LAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_LAR	R	0h	Lock status

21.2.81 ONEMCU_TPIU_LSR Register (Offset = 2FB4h) [Reset = 00000000h]

ONEMCU_TPIU_LSR is shown in Table 21-84.

Return to the Summary Table.

Lock Access

Table 21-84. ONEMCU_TPIU_LSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_LSR	W	0h	Lock Access

21.2.82 ONEMCU_TPIU_AUTHSTATUS Register (Offset = 2FB8h) [Reset = 00000000h]

ONEMCU_TPIU_AUTHSTATUS is shown in Table 21-85.

Return to the Summary Table.

Authentication status

Table 21-85. ONEMCU_TPIU_AUTHSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_AUTHSTATUS	R	00h	Authentication status

21.2.83 ONEMCU_TPIU_DEVID Register (Offset = 2FC8h) [Reset = 000000A0h]

ONEMCU_TPIU_DEVID is shown in Table 21-86.

Return to the Summary Table.

Device ID

Table 21-86. ONEMCU_TPIU_DEVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_DEVID	R	0A0h	Device ID

21.2.84 ONEMCU_TPIU_DEVTYPE Register (Offset = 2FCCh) [Reset = 00000011h]

ONEMCU_TPIU_DEVTYPE is shown in Table 21-87.

Return to the Summary Table.

Device type identifier

Table 21-87. ONEMCU_TPIU_DEVTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_DEVTYPE	R	11h	Device type identifier

21.2.85 ONEMCU_TPIU_PIDR4 Register (Offset = 2FD0h) [Reset = 00000004h]

ONEMCU_TPIU_PIDR4 is shown in Table 21-88.

Return to the Summary Table.

Peripheral ID4

Table 21-88. ONEMCU_TPIU_PIDR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR4	R	04h	Peripheral ID4

21.2.86 ONEMCU_TPIU_PIDR5 Register (Offset = 2FD4h) [Reset = 00000000h]

ONEMCU_TPIU_PIDR5 is shown in Table 21-89.

Return to the Summary Table.

Peripheral ID5

Table 21-89. ONEMCU_TPIU_PIDR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR5	R	0h	Peripheral ID5

21.2.87 ONEMCU_TPIU_PIDR6 Register (Offset = 2FD8h) [Reset = 00000000h]

ONEMCU_TPIU_PIDR6 is shown in Table 21-90.

Return to the Summary Table.

Peripheral ID6

Table 21-90. ONEMCU_TPIU_PIDR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR6	R	0h	Peripheral ID6

21.2.88 ONEMCU_TPIU_PIDR7 Register (Offset = 2FDCh) [Reset = 00000000h]

ONEMCU_TPIU_PIDR7 is shown in Table 21-91.

Return to the Summary Table.

Peripheral ID7

Table 21-91. ONEMCU_TPIU_PIDR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR7	R	0h	Peripheral ID7

21.2.89 ONEMCU_TPIU_PIDR0 Register (Offset = 2FE0h) [Reset = 00000012h]

ONEMCU_TPIU_PIDR0 is shown in Table 21-92.

Return to the Summary Table.

Peripheral ID0

Table 21-92. ONEMCU_TPIU_PIDR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR0	R	12h	Peripheral ID0

21.2.90 ONEMCU_TPIU_PIDR1 Register (Offset = 2FE4h) [Reset = 000000B9h]

ONEMCU_TPIU_PIDR1 is shown in Table 21-93.

Return to the Summary Table.

Peripheral ID1

Table 21-93. ONEMCU_TPIU_PIDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR1	R	B9h	Peripheral ID1

21.2.91 ONEMCU_TPIU_PIDR2 Register (Offset = 2FE8h) [Reset = 0000003Bh]

ONEMCU_TPIU_PIDR2 is shown in Table 21-94.

Return to the Summary Table.

Peripheral ID2

Table 21-94. ONEMCU_TPIU_PIDR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR2	R	3Bh	Peripheral ID2

21.2.92 ONEMCU_TPIU_PIDR3 Register (Offset = 2FECh) [Reset = 00000000h]

ONEMCU_TPIU_PIDR3 is shown in Table 21-95.

Return to the Summary Table.

Peripheral ID3

Table 21-95. ONEMCU_TPIU_PIDR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_PIDR3	R	00h	Peripheral ID3

21.2.93 ONEMCU_TPIU_CIDR0 Register (Offset = 2FF0h) [Reset = 0000000Dh]

ONEMCU_TPIU_CIDR0 is shown in Table 21-96.

Return to the Summary Table.

Component ID0

Table 21-96. ONEMCU_TPIU_CIDR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CIDR0	R	0Dh	Component ID0

21.2.94 ONEMCU_TPIU_CIDR1 Register (Offset = 2FF4h) [Reset = 00000090h]

ONEMCU_TPIU_CIDR1 is shown in Table 21-97.

Return to the Summary Table.

Component ID1

Table 21-97. ONEMCU_TPIU_CIDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CIDR1	R	90h	Component ID1

21.2.95 ONEMCU_TPIU_CIDR2 Register (Offset = 2FF8h) [Reset = 00000005h]

ONEMCU_TPIU_CIDR2 is shown in Table 21-98.

Return to the Summary Table.

Component ID2

Table 21-98. ONEMCU_TPIU_CIDR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CIDR2	R	05h	Component ID2

21.2.96 ONEMCU_TPIU_CIDR3 Register (Offset = 2FFCh) [Reset = 000000B1h]

ONEMCU_TPIU_CIDR3 is shown in Table 21-99.

Return to the Summary Table.

Component ID3

Table 21-99. ONEMCU_TPIU_CIDR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ONEMCU_TPIU_CIDR3	R	B1h	Component ID3

21.2.97 APP_CM4_CTI_CONTROL Register (Offset = 00010000h) [Reset = 00000000h]

APP_CM4_CTI_CONTROL is shown in Table 21-100.

Return to the Summary Table.

http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdjefbi.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdefejc.html

Table 21-100. APP_CM4_CTI_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_CONTROL	R/W	0h	http://infocenter.arm.com/help/topic/com.arm.doc.ddi0480e/CHDGDIHE.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0480e/CHDHBDIA.html

21.2.98 APP_CM4_CTI_INTACK Register (Offset = 00010010h) [Reset = 00000000h]

APP_CM4_CTI_INTACK is shown in Table 21-101.

Return to the Summary Table.

Table 21-101. APP_CM4_CTI_INTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INTACK	W	0h	

21.2.99 APP_CM4_CTI_APPSET Register (Offset = 00010014h) [Reset = 00000000h]

APP_CM4_CTI_APPSET is shown in Table 21-102.

Return to the Summary Table.

Table 21-102. APP_CM4_CTI_APPSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_APPSET	R/W	0h	

21.2.100 APP_CM4_CTI_APPCLEAR Register (Offset = 00010018h) [Reset = 00000000h]

APP_CM4_CTI_APPCLEAR is shown in Table 21-103.

Return to the Summary Table.

Table 21-103. APP_CM4_CTI_APPCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_APPCLEAR	W	0h	

21.2.101 APP_CM4_CTI_APPPULSE Register (Offset = 0001001Ch) [Reset = 00000000h]

APP_CM4_CTI_APPPULSE is shown in Table 21-104.

Return to the Summary Table.

Table 21-104. APP_CM4_CTI_APPPULSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_APPPULSE	W	0h	

21.2.102 APP_CM4_CTI_INEN0 Register (Offset = 00010020h) [Reset = 00000000h]

APP_CM4_CTI_INEN0 is shown in Table 21-105.

Return to the Summary Table.

Table 21-105. APP_CM4_CTI_INEN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN0	R/W	0h	

21.2.103 APP_CM4_CTI_INEN1 Register (Offset = 00010024h) [Reset = 00000000h]

APP_CM4_CTI_INEN1 is shown in Table 21-106.

Return to the Summary Table.

Table 21-106. APP_CM4_CTI_INEN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN1	R/W	0h	

21.2.104 APP_CM4_CTI_INEN2 Register (Offset = 00010028h) [Reset = 00000000h]

APP_CM4_CTI_INEN2 is shown in Table 21-107.

Return to the Summary Table.

Table 21-107. APP_CM4_CTI_INEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN2	R/W	0h	

21.2.105 APP_CM4_CTI_INEN3 Register (Offset = 0001002Ch) [Reset = 00000000h]

APP_CM4_CTI_INEN3 is shown in Table 21-108.

Return to the Summary Table.

Table 21-108. APP_CM4_CTI_INEN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN3	R/W	0h	

21.2.106 APP_CM4_CTI_INEN4 Register (Offset = 00010030h) [Reset = 00000000h]

APP_CM4_CTI_INEN4 is shown in Table 21-109.

Return to the Summary Table.

Table 21-109. APP_CM4_CTI_INEN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN4	R/W	0h	

21.2.107 APP_CM4_CTI_INEN5 Register (Offset = 00010034h) [Reset = 00000000h]

APP_CM4_CTI_INEN5 is shown in Table 21-110.

Return to the Summary Table.

Table 21-110. APP_CM4_CTI_INEN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN5	R/W	0h	

21.2.108 APP_CM4_CTI_INEN6 Register (Offset = 00010038h) [Reset = 00000000h]

APP_CM4_CTI_INEN6 is shown in Table 21-111.

Return to the Summary Table.

Table 21-111. APP_CM4_CTI_INEN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN6	R/W	0h	

21.2.109 APP_CM4_CTI_INEN7 Register (Offset = 0001003Ch) [Reset = 00000000h]

APP_CM4_CTI_INEN7 is shown in Table 21-112.

Return to the Summary Table.

Table 21-112. APP_CM4_CTI_INEN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_INEN7	R/W	0h	

21.2.110 APP_CM4_CTI_OUTEN0 Register (Offset = 000100A0h) [Reset = 00000000h]

APP_CM4_CTI_OUTEN0 is shown in Table 21-113.

Return to the Summary Table.

Table 21-113. APP_CM4_CTI_OUTEN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN0	R/W	0h	

21.2.111 APP_CM4_CTI_OUTEN1 Register (Offset = 000100A4h) [Reset = 00000000h]

APP_CM4_CTI_OUTEN1 is shown in Table 21-114.

Return to the Summary Table.

Table 21-114. APP_CM4_CTI_OUTEN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN1	R/W	0h	

21.2.112 APP_CM4_CTI_OUTEN2 Register (Offset = 000100A8h) [Reset = 00000000h]

APP_CM4_CTI_OUTEN2 is shown in Table 21-115.

Return to the Summary Table.

Table 21-115. APP_CM4_CTI_OUTEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN2	R/W	0h	

21.2.113 APP_CM4_CTI_OUTEN3 Register (Offset = 000100ACh) [Reset = 00000000h]

APP_CM4_CTI_OUTEN3 is shown in Table 21-116.

Return to the Summary Table.

Table 21-116. APP_CM4_CTI_OUTEN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN3	R/W	0h	

21.2.114 APP_CM4_CTI_OUTEN4 Register (Offset = 000100B0h) [Reset = 00000000h]

APP_CM4_CTI_OUTEN4 is shown in Table 21-117.

Return to the Summary Table.

Table 21-117. APP_CM4_CTI_OUTEN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN4	R/W	0h	

21.2.115 APP_CM4_CTI_OUTEN5 Register (Offset = 000100B4h) [Reset = 00000000h]

APP_CM4_CTI_OUTEN5 is shown in Table 21-118.

Return to the Summary Table.

Table 21-118. APP_CM4_CTI_OUTEN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN5	R/W	0h	

21.2.116 APP_CM4_CTI_OUTEN6 Register (Offset = 000100B8h) [Reset = 00000000h]

APP_CM4_CTI_OUTEN6 is shown in Table 21-119.

Return to the Summary Table.

Table 21-119. APP_CM4_CTI_OUTEN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN6	R/W	0h	

21.2.117 APP_CM4_CTI_OUTEN7 Register (Offset = 000100BCh) [Reset = 00000000h]

APP_CM4_CTI_OUTEN7 is shown in Table 21-120.

Return to the Summary Table.

Table 21-120. APP_CM4_CTI_OUTEN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_OUTEN7	R/W	0h	

21.2.118 APP_CM4_CTI_TRIGINSTATUS Register (Offset = 00010130h) [Reset = 00000000h]

APP_CM4_CTI_TRIGINSTATUS is shown in Table 21-121.

Return to the Summary Table.

Table 21-121. APP_CM4_CTI_TRIGINSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_TRIGINSTATUS	R	0h	

21.2.119 APP_CM4_CTI_TRIGOUTSTATUS Register (Offset = 00010134h) [Reset = 00000000h]

APP_CM4_CTI_TRIGOUTSTATUS is shown in Table 21-122.

Return to the Summary Table.

Table 21-122. APP_CM4_CTI_TRIGOUTSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_TRIGOUTSTATUS	R	0h	

21.2.120 APP_CM4_CTI_CHINSTATUS Register (Offset = 00010138h) [Reset = 00000000h]

APP_CM4_CTI_CHINSTATUS is shown in Table 21-123.

Return to the Summary Table.

Table 21-123. APP_CM4_CTI_CHINSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_CHINSTATUS	R	0h	

21.2.121 APP_CM4_CTI_CHOUTSTATUS Register (Offset = 0001013Ch) [Reset = 00000000h]

APP_CM4_CTI_CHOUTSTATUS is shown in Table 21-124.

Return to the Summary Table.

Table 21-124. APP_CM4_CTI_CHOUTSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_CHOUTSTATUS	R	0h	

21.2.122 APP_CM4_CTI_GATE Register (Offset = 00010140h) [Reset = 00000000h]

APP_CM4_CTI_GATE is shown in Table 21-125.

Return to the Summary Table.

Table 21-125. APP_CM4_CTI_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_GATE	R/W	0h	

21.2.123 APP_CM4_CTI_ASICCTL Register (Offset = 00010144h) [Reset = 00000000h]

APP_CM4_CTI_ASICCTL is shown in Table 21-126.

Return to the Summary Table.

Table 21-126. APP_CM4_CTI_ASICCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ASICCTL	R/W	0h	

21.2.124 APP_CM4_CTI_ITCHINACK Register (Offset = 00010EDCh) [Reset = 00000000h]

APP_CM4_CTI_ITCHINACK is shown in Table 21-127.

Return to the Summary Table.

Table 21-127. APP_CM4_CTI_ITCHINACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITCHINACK	W	0h	

21.2.125 APP_CM4_CTI_ITTRIGINACK Register (Offset = 00010EE0h) [Reset = 00000000h]

APP_CM4_CTI_ITTRIGINACK is shown in Table 21-128.

Return to the Summary Table.

Table 21-128. APP_CM4_CTI_ITTRIGINACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITTRIGINACK	W	0h	

21.2.126 APP_CM4_CTI_ITCHOUT Register (Offset = 00010EE4h) [Reset = 00000000h]

APP_CM4_CTI_ITCHOUT is shown in Table 21-129.

Return to the Summary Table.

Table 21-129. APP_CM4_CTI_ITCHOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITCHOUT	W	0h	

21.2.127 APP_CM4_CTI_ITTRIGOUT Register (Offset = 00010EE8h) [Reset = 00000000h]

APP_CM4_CTI_ITTRIGOUT is shown in Table 21-130.

Return to the Summary Table.

Table 21-130. APP_CM4_CTI_ITTRIGOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITTRIGOUT	W	0h	

21.2.128 APP_CM4_CTI_ITCHOUTACK Register (Offset = 00010EECh) [Reset = 00000000h]

APP_CM4_CTI_ITCHOUTACK is shown in Table 21-131.

Return to the Summary Table.

Table 21-131. APP_CM4_CTI_ITCHOUTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITCHOUTACK	R	0h	

21.2.129 APP_CM4_CTI_ITTRIGOUTACK Register (Offset = 00010EF0h) [Reset = 00000000h]

APP_CM4_CTI_ITTRIGOUTACK is shown in Table 21-132.

Return to the Summary Table.

Table 21-132. APP_CM4_CTI_ITTRIGOUTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITTRIGOUTACK	R	0h	

21.2.130 APP_CM4_CTI_ITCHIN Register (Offset = 00010EF4h) [Reset = 00000000h]

APP_CM4_CTI_ITCHIN is shown in Table 21-133.

Return to the Summary Table.

Table 21-133. APP_CM4_CTI_ITCHIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITCHIN	R	0h	

21.2.131 APP_CM4_CTI_ITTRIGIN Register (Offset = 00010EF8h) [Reset = 00000000h]

APP_CM4_CTI_ITTRIGIN is shown in Table 21-134.

Return to the Summary Table.

Table 21-134. APP_CM4_CTI_ITTRIGIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITTRIGIN	R	0h	

21.2.132 APP_CM4_CTI_ITCTRL Register (Offset = 00010F00h) [Reset = 00000000h]

APP_CM4_CTI_ITCTRL is shown in Table 21-135.

Return to the Summary Table.

Table 21-135. APP_CM4_CTI_ITCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_ITCTRL	R/W	0h	

21.2.133 APP_CM4_CTI_Claim_Tag_Set Register (Offset = 00010FA0h) [Reset = 00000000h]

APP_CM4_CTI_Claim_Tag_Set is shown in Table 21-136.

Return to the Summary Table.

Table 21-136. APP_CM4_CTI_Claim_Tag_Set Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Claim_Tag_Set	R/W	0h	

21.2.134 APP_CM4_CTI_Claim_Tag_Clear Register (Offset = 00010FA4h) [Reset = 00000000h]

APP_CM4_CTI_Claim_Tag_Clear is shown in Table 21-137.

Return to the Summary Table.

Table 21-137. APP_CM4_CTI_Claim_Tag_Clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Claim_Tag_Clear	R/W	0h	

21.2.135 APP_CM4_CTI_Lock_Access_Register (Offset = 00010FB0h) [Reset = 00000000h]

APP_CM4_CTI_Lock_Access_Register is shown in Table 21-138.

Return to the Summary Table.

Table 21-138. APP_CM4_CTI_Lock_Access_Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Lock_Access_Register	W	0h	

21.2.136 APP_CM4_CTI_Lock_Status_Register (Offset = 00010FB4h) [Reset = 00000000h]

APP_CM4_CTI_Lock_Status_Register is shown in Table 21-139.

Return to the Summary Table.

Table 21-139. APP_CM4_CTI_Lock_Status_Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Lock_Status_Register	R	0h	

21.2.137 APP_CM4_CTI_Authentication_Status Register (Offset = 00010FB8h) [Reset = 00000000h]

APP_CM4_CTI_Authentication_Status is shown in Table 21-140.

Return to the Summary Table.

Table 21-140. APP_CM4_CTI_Authentication_Status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Authentication_Status	R	0h	

21.2.138 APP_CM4_CTI_Device_ID Register (Offset = 00010FC8h) [Reset = 00000000h]

APP_CM4_CTI_Device_ID is shown in Table 21-141.

Return to the Summary Table.

Table 21-141. APP_CM4_CTI_Device_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Device_ID	R	0h	

21.2.139 APP_CM4_CTI_Device_Type_Identifier Register (Offset = 00010FCCh) [Reset = 00000000h]

APP_CM4_CTI_Device_Type_Identifier is shown in Table 21-142.

Return to the Summary Table.

Table 21-142. APP_CM4_CTI_Device_Type_Identifier Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Device_Type_Identifier	R	0h	

21.2.140 APP_CM4_CTI_PeripheralID4 Register (Offset = 00010FD0h) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID4 is shown in Table 21-143.

Return to the Summary Table.

Table 21-143. APP_CM4_CTI_PeripheralID4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID4	R	0h	

21.2.141 APP_CM4_CTI_PeripheralID5 Register (Offset = 00010FD4h) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID5 is shown in Table 21-144.

Return to the Summary Table.

Table 21-144. APP_CM4_CTI_PeripheralID5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID5	R	0h	

21.2.142 APP_CM4_CTI_PeripheralID6 Register (Offset = 00010FD8h) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID6 is shown in Table 21-145.

Return to the Summary Table.

Table 21-145. APP_CM4_CTI_PeripheralID6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID6	R	0h	

21.2.143 APP_CM4_CTI_PeripheralID7 Register (Offset = 00010FDCh) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID7 is shown in Table 21-146.

Return to the Summary Table.

Table 21-146. APP_CM4_CTI_PeripheralID7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID7	R	0h	

21.2.144 APP_CM4_CTI_PeripheralID0 Register (Offset = 00010FE0h) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID0 is shown in Table 21-147.

Return to the Summary Table.

Table 21-147. APP_CM4_CTI_PeripheralID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID0	R	0h	

21.2.145 APP_CM4_CTI_PeripheralID1 Register (Offset = 00010FE4h) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID1 is shown in Table 21-148.

Return to the Summary Table.

Table 21-148. APP_CM4_CTI_PeripheralID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID1	R	0h	

21.2.146 APP_CM4_CTI_PeripheralID2 Register (Offset = 00010FE8h) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID2 is shown in Table 21-149.

Return to the Summary Table.

Table 21-149. APP_CM4_CTI_PeripheralID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID2	R	0h	

21.2.147 APP_CM4_CTI_PeripheralID3 Register (Offset = 00010FECh) [Reset = 00000000h]

APP_CM4_CTI_PeripheralID3 is shown in Table 21-150.

Return to the Summary Table.

Table 21-150. APP_CM4_CTI_PeripheralID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_PeripheralID3	R	0h	

21.2.148 APP_CM4_CTI_Component_ID0 Register (Offset = 00010FF0h) [Reset = 00000000h]

APP_CM4_CTI_Component_ID0 is shown in Table 21-151.

Return to the Summary Table.

Table 21-151. APP_CM4_CTI_Component_ID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Component_ID0	R	0h	

21.2.149 APP_CM4_CTI_Component_ID1 Register (Offset = 00010FF4h) [Reset = 00000000h]

APP_CM4_CTI_Component_ID1 is shown in Table 21-152.

Return to the Summary Table.

Table 21-152. APP_CM4_CTI_Component_ID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Component_ID1	R	0h	

21.2.150 APP_CM4_CTI_Component_ID2 Register (Offset = 00010FF8h) [Reset = 00000000h]

APP_CM4_CTI_Component_ID2 is shown in Table 21-153.

Return to the Summary Table.

Table 21-153. APP_CM4_CTI_Component_ID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Component_ID2	R	0h	

21.2.151 APP_CM4_CTI_Component_ID3 Register (Offset = 00010FFCh) [Reset = 00000000h]

APP_CM4_CTI_Component_ID3 is shown in Table 21-154.

Return to the Summary Table.

Table 21-154. APP_CM4_CTI_Component_ID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_CM4_CTI_Component_ID3	R	0h	

21.2.152 FEC_CM3_CTI_CONTROL Register (Offset = 00011000h) [Reset = 00000000h]

FEC_CM3_CTI_CONTROL is shown in Table 21-155.

Return to the Summary Table.

http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdjefbi.html http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdefejc.html

Table 21-155. FEC_CM3_CTI_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_CONTROL	R/W	0h	

21.2.153 FEC_CM3_CTI_INTACK Register (Offset = 00011010h) [Reset = 00000000h]

FEC_CM3_CTI_INTACK is shown in Table 21-156.

Return to the Summary Table.

Table 21-156. FEC_CM3_CTI_INTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INTACK	W	0h	

21.2.154 FEC_CM3_CTI_APPSET Register (Offset = 00011014h) [Reset = 00000000h]

FEC_CM3_CTI_APPSET is shown in Table 21-157.

Return to the Summary Table.

Table 21-157. FEC_CM3_CTI_APPSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_APPSET	R/W	0h	

21.2.155 FEC_CM3_CTI_APPCLEAR Register (Offset = 00011018h) [Reset = 00000000h]

FEC_CM3_CTI_APPCLEAR is shown in Table 21-158.

Return to the Summary Table.

Table 21-158. FEC_CM3_CTI_APPCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_APPCLEAR	W	0h	

21.2.156 FEC_CM3_CTI_APPPULSE Register (Offset = 0001101Ch) [Reset = 00000000h]

FEC_CM3_CTI_APPPULSE is shown in Table 21-159.

Return to the Summary Table.

Table 21-159. FEC_CM3_CTI_APPPULSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_APPPULSE	W	0h	

21.2.157 FEC_CM3_CTI_INEN0 Register (Offset = 00011020h) [Reset = 00000000h]

FEC_CM3_CTI_INEN0 is shown in Table 21-160.

Return to the Summary Table.

Table 21-160. FEC_CM3_CTI_INEN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN0	R/W	0h	

21.2.158 FEC_CM3_CTI_INEN1 Register (Offset = 00011024h) [Reset = 00000000h]

FEC_CM3_CTI_INEN1 is shown in Table 21-161.

Return to the Summary Table.

Table 21-161. FEC_CM3_CTI_INEN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN1	R/W	0h	

21.2.159 FEC_CM3_CTI_INEN2 Register (Offset = 00011028h) [Reset = 00000000h]

FEC_CM3_CTI_INEN2 is shown in Table 21-162.

Return to the Summary Table.

Table 21-162. FEC_CM3_CTI_INEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN2	R/W	0h	

21.2.160 FEC_CM3_CTI_INEN3 Register (Offset = 0001102Ch) [Reset = 00000000h]

FEC_CM3_CTI_INEN3 is shown in Table 21-163.

Return to the Summary Table.

Table 21-163. FEC_CM3_CTI_INEN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN3	R/W	0h	

21.2.161 FEC_CM3_CTI_INEN4 Register (Offset = 00011030h) [Reset = 00000000h]

FEC_CM3_CTI_INEN4 is shown in Table 21-164.

Return to the Summary Table.

Table 21-164. FEC_CM3_CTI_INEN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN4	R/W	0h	

21.2.162 FEC_CM3_CTI_INEN5 Register (Offset = 00011034h) [Reset = 00000000h]

FEC_CM3_CTI_INEN5 is shown in Table 21-165.

Return to the Summary Table.

Table 21-165. FEC_CM3_CTI_INEN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN5	R/W	0h	

21.2.163 FEC_CM3_CTI_INEN6 Register (Offset = 00011038h) [Reset = 00000000h]

FEC_CM3_CTI_INEN6 is shown in Table 21-166.

Return to the Summary Table.

Table 21-166. FEC_CM3_CTI_INEN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN6	R/W	0h	

21.2.164 FEC_CM3_CTI_INEN7 Register (Offset = 0001103Ch) [Reset = 00000000h]

FEC_CM3_CTI_INEN7 is shown in Table 21-167.

Return to the Summary Table.

Table 21-167. FEC_CM3_CTI_INEN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_INEN7	R/W	0h	

21.2.165 FEC_CM3_CTI_OUTEN0 Register (Offset = 000110A0h) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN0 is shown in Table 21-168.

Return to the Summary Table.

Table 21-168. FEC_CM3_CTI_OUTEN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN0	R/W	0h	

21.2.166 FEC_CM3_CTI_OUTEN1 Register (Offset = 000110A4h) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN1 is shown in Table 21-169.

Return to the Summary Table.

Table 21-169. FEC_CM3_CTI_OUTEN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN1	R/W	0h	

21.2.167 FEC_CM3_CTI_OUTEN2 Register (Offset = 000110A8h) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN2 is shown in Table 21-170.

Return to the Summary Table.

Table 21-170. FEC_CM3_CTI_OUTEN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN2	R/W	0h	

21.2.168 FEC_CM3_CTI_OUTEN3 Register (Offset = 000110ACh) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN3 is shown in Table 21-171.

Return to the Summary Table.

Table 21-171. FEC_CM3_CTI_OUTEN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN3	R/W	0h	

21.2.169 FEC_CM3_CTI_OUTEN4 Register (Offset = 000110B0h) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN4 is shown in Table 21-172.

Return to the Summary Table.

Table 21-172. FEC_CM3_CTI_OUTEN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN4	R/W	0h	

21.2.170 FEC_CM3_CTI_OUTEN5 Register (Offset = 000110B4h) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN5 is shown in Table 21-173.

Return to the Summary Table.

Table 21-173. FEC_CM3_CTI_OUTEN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN5	R/W	0h	

21.2.171 FEC_CM3_CTI_OUTEN6 Register (Offset = 000110B8h) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN6 is shown in Table 21-174.

Return to the Summary Table.

Table 21-174. FEC_CM3_CTI_OUTEN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN6	R/W	0h	

21.2.172 FEC_CM3_CTI_OUTEN7 Register (Offset = 000110BCh) [Reset = 00000000h]

FEC_CM3_CTI_OUTEN7 is shown in Table 21-175.

Return to the Summary Table.

Table 21-175. FEC_CM3_CTI_OUTEN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_OUTEN7	R/W	0h	

21.2.173 FEC_CM3_CTI_TRIGINSTATUS Register (Offset = 00011130h) [Reset = 00000000h]

FEC_CM3_CTI_TRIGINSTATUS is shown in Table 21-176.

Return to the Summary Table.

Table 21-176. FEC_CM3_CTI_TRIGINSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_TRIGINSTATUS	R	0h	

21.2.174 FEC_CM3_CTI_TRIGOUTSTATUS Register (Offset = 00011134h) [Reset = 00000000h]

FEC_CM3_CTI_TRIGOUTSTATUS is shown in Table 21-177.

Return to the Summary Table.

Table 21-177. FEC_CM3_CTI_TRIGOUTSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_TRIGOUTSTATUS	R	0h	

21.2.175 FEC_CM3_CTI_CHINSTATUS Register (Offset = 00011138h) [Reset = 00000000h]

FEC_CM3_CTI_CHINSTATUS is shown in Table 21-178.

Return to the Summary Table.

Table 21-178. FEC_CM3_CTI_CHINSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_CHINSTATUS	R	0h	

21.2.176 FEC_CM3_CTI_CHOUTSTATUS Register (Offset = 0001113Ch) [Reset = 00000000h]

FEC_CM3_CTI_CHOUTSTATUS is shown in Table 21-179.

Return to the Summary Table.

Table 21-179. FEC_CM3_CTI_CHOUTSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_CHOUTSTATUS	R	0h	

21.2.177 FEC_CM3_CTI_GATE Register (Offset = 00011140h) [Reset = 00000000h]

FEC_CM3_CTI_GATE is shown in Table 21-180.

Return to the Summary Table.

Table 21-180. FEC_CM3_CTI_GATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_GATE	R/W	0h	

21.2.178 FEC_CM3_CTI_ASICCTL Register (Offset = 00011144h) [Reset = 00000000h]

FEC_CM3_CTI_ASICCTL is shown in Table 21-181.

Return to the Summary Table.

Table 21-181. FEC_CM3_CTI_ASICCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ASICCTL	R/W	0h	

21.2.179 FEC_CM3_CTI_ITCHINACK Register (Offset = 00011EDCh) [Reset = 00000000h]

FEC_CM3_CTI_ITCHINACK is shown in Table 21-182.

Return to the Summary Table.

Table 21-182. FEC_CM3_CTI_ITCHINACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITCHINACK	W	0h	

21.2.180 FEC_CM3_CTI_ITTRIGINACK Register (Offset = 00011EE0h) [Reset = 00000000h]

FEC_CM3_CTI_ITTRIGINACK is shown in Table 21-183.

Return to the Summary Table.

Table 21-183. FEC_CM3_CTI_ITTRIGINACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITTRIGINACK	W	0h	

21.2.181 FEC_CM3_CTI_ITCHOUT Register (Offset = 00011EE4h) [Reset = 00000000h]

FEC_CM3_CTI_ITCHOUT is shown in Table 21-184.

Return to the Summary Table.

Table 21-184. FEC_CM3_CTI_ITCHOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITCHOUT	W	0h	

21.2.182 FEC_CM3_CTI_ITTRIGOUT Register (Offset = 00011EE8h) [Reset = 00000000h]

FEC_CM3_CTI_ITTRIGOUT is shown in Table 21-185.

Return to the Summary Table.

Table 21-185. FEC_CM3_CTI_ITTRIGOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITTRIGOUT	W	0h	

21.2.183 FEC_CM3_CTI_ITCHOUTACK Register (Offset = 00011EECh) [Reset = 00000000h]

FEC_CM3_CTI_ITCHOUTACK is shown in Table 21-186.

Return to the Summary Table.

Table 21-186. FEC_CM3_CTI_ITCHOUTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITCHOUTACK	R	0h	

21.2.184 FEC_CM3_CTI_ITTRIGOUTACK Register (Offset = 00011EF0h) [Reset = 00000000h]

FEC_CM3_CTI_ITTRIGOUTACK is shown in Table 21-187.

Return to the Summary Table.

Table 21-187. FEC_CM3_CTI_ITTRIGOUTACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITTRIGOUTACK	R	0h	

21.2.185 FEC_CM3_CTI_ITCHIN Register (Offset = 00011EF4h) [Reset = 00000000h]

FEC_CM3_CTI_ITCHIN is shown in Table 21-188.

Return to the Summary Table.

Table 21-188. FEC_CM3_CTI_ITCHIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITCHIN	R	0h	

21.2.186 FEC_CM3_CTI_ITTRIGIN Register (Offset = 00011EF8h) [Reset = 00000000h]

FEC_CM3_CTI_ITTRIGIN is shown in Table 21-189.

Return to the Summary Table.

Table 21-189. FEC_CM3_CTI_ITTRIGIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITTRIGIN	R	0h	

21.2.187 FEC_CM3_CTI_ITCTRL Register (Offset = 00011F00h) [Reset = 00000000h]

FEC_CM3_CTI_ITCTRL is shown in Table 21-190.

Return to the Summary Table.

Table 21-190. FEC_CM3_CTI_ITCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_ITCTRL	R/W	0h	

21.2.188 FEC_CM3_CTI_Claim_Tag_Set Register (Offset = 00011FA0h) [Reset = 00000000h]

FEC_CM3_CTI_Claim_Tag_Set is shown in Table 21-191.

Return to the Summary Table.

Table 21-191. FEC_CM3_CTI_Claim_Tag_Set Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Claim_Tag_Set	R/W	0h	

21.2.189 FEC_CM3_CTI_Claim_Tag_Clear Register (Offset = 00011FA4h) [Reset = 00000000h]

FEC_CM3_CTI_Claim_Tag_Clear is shown in Table 21-192.

Return to the Summary Table.

Table 21-192. FEC_CM3_CTI_Claim_Tag_Clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Claim_Tag_Clear	R/W	0h	

21.2.190 FEC_CM3_CTI_Lock_Access_Register (Offset = 00011FB0h) [Reset = 00000000h]

FEC_CM3_CTI_Lock_Access_Register is shown in Table 21-193.

Return to the Summary Table.

Table 21-193. FEC_CM3_CTI_Lock_Access_Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Lock_Access_Register	W	0h	

21.2.191 FEC_CM3_CTI_Lock_Status_Register (Offset = 00011FB4h) [Reset = 00000000h]

FEC_CM3_CTI_Lock_Status_Register is shown in Table 21-194.

Return to the Summary Table.

Table 21-194. FEC_CM3_CTI_Lock_Status_Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Lock_Status_Register	R	0h	

21.2.192 FEC_CM3_CTI_Authentication_Status Register (Offset = 00011FB8h) [Reset = 00000000h]

FEC_CM3_CTI_Authentication_Status is shown in Table 21-195.

Return to the Summary Table.

Table 21-195. FEC_CM3_CTI_Authentication_Status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Authentication_Status	R	0h	

21.2.193 FEC_CM3_CTI_Device_ID Register (Offset = 00011FC8h) [Reset = 00000000h]

FEC_CM3_CTI_Device_ID is shown in Table 21-196.

Return to the Summary Table.

Table 21-196. FEC_CM3_CTI_Device_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Device_ID	R	0h	

21.2.194 FEC_CM3_CTI_Device_Type_Identifier Register (Offset = 00011FCCh) [Reset = 00000000h]

FEC_CM3_CTI_Device_Type_Identifier is shown in Table 21-197.

Return to the Summary Table.

Table 21-197. FEC_CM3_CTI_Device_Type_Identifier Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Device_Type_Identifier	R	0h	

21.2.195 FEC_CM3_CTI_PeripheralID4 Register (Offset = 00011FD0h) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID4 is shown in Table 21-198.

Return to the Summary Table.

Table 21-198. FEC_CM3_CTI_PeripheralID4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID4	R	0h	

21.2.196 FEC_CM3_CTI_PeripheralID5 Register (Offset = 00011FD4h) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID5 is shown in Table 21-199.

Return to the Summary Table.

Table 21-199. FEC_CM3_CTI_PeripheralID5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID5	R	0h	

21.2.197 FEC_CM3_CTI_PeripheralID6 Register (Offset = 00011FD8h) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID6 is shown in Table 21-200.

Return to the Summary Table.

Table 21-200. FEC_CM3_CTI_PeripheralID6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID6	R	0h	

21.2.198 FEC_CM3_CTI_PeripheralID7 Register (Offset = 00011FDCh) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID7 is shown in Table 21-201.

Return to the Summary Table.

Table 21-201. FEC_CM3_CTI_PeripheralID7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID7	R	0h	

21.2.199 FEC_CM3_CTI_PeripheralID0 Register (Offset = 00011FE0h) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID0 is shown in Table 21-202.

Return to the Summary Table.

Table 21-202. FEC_CM3_CTI_PeripheralID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID0	R	0h	

21.2.200 FEC_CM3_CTI_PeripheralID1 Register (Offset = 00011FE4h) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID1 is shown in Table 21-203.

Return to the Summary Table.

Table 21-203. FEC_CM3_CTI_PeripheralID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID1	R	0h	

21.2.201 FEC_CM3_CTI_PeripheralID2 Register (Offset = 00011FE8h) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID2 is shown in Table 21-204.

Return to the Summary Table.

Table 21-204. FEC_CM3_CTI_PeripheralID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID2	R	0h	

21.2.202 FEC_CM3_CTI_PeripheralID3 Register (Offset = 00011FECh) [Reset = 00000000h]

FEC_CM3_CTI_PeripheralID3 is shown in Table 21-205.

Return to the Summary Table.

Table 21-205. FEC_CM3_CTI_PeripheralID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_PeripheralID3	R	0h	

21.2.203 FEC_CM3_CTI_Component_ID0 Register (Offset = 00011FF0h) [Reset = 00000000h]

FEC_CM3_CTI_Component_ID0 is shown in Table 21-206.

Return to the Summary Table.

Table 21-206. FEC_CM3_CTI_Component_ID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Component_ID0	R	0h	

21.2.204 FEC_CM3_CTI_Component_ID1 Register (Offset = 00011FF4h) [Reset = 00000000h]

FEC_CM3_CTI_Component_ID1 is shown in Table 21-207.

Return to the Summary Table.

Table 21-207. FEC_CM3_CTI_Component_ID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Component_ID1	R	0h	

21.2.205 FEC_CM3_CTI_Component_ID2 Register (Offset = 00011FF8h) [Reset = 00000000h]

FEC_CM3_CTI_Component_ID2 is shown in Table 21-208.

Return to the Summary Table.

Table 21-208. FEC_CM3_CTI_Component_ID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Component_ID2	R	0h	

21.2.206 FEC_CM3_CTI_Component_ID3 Register (Offset = 00011FFCh) [Reset = 00000000h]

FEC_CM3_CTI_Component_ID3 is shown in Table 21-209.

Return to the Summary Table.

Table 21-209. FEC_CM3_CTI_Component_ID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FEC_CM3_CTI_Component_ID3	R	0h	

22 Safety Modules

22.1 Logic Self-Test Controller

Design and Microarchitecture Specification

22.1.1 LBIST Self-Test Controller Overview

22.1.1.1 Scope

This document describes the design and micro-architecture details of the LSTC module.

22.1.1.2 General Description

LSTC stands for LBIST Self-Test Controller. LBIST is a design for testability (DFT) technique which uses on-chip circuitry to generate test patterns and analyze test responses on-chip without ATE intervention. LBIST Self-Test Controller (LSTC) IP addresses the logic self-test requirements for functional safety applications as per ISO-26262 ASIL compliance. The LBIST is triggered through software during power-up and/or during functional operation. LSTC IP consists of sequencer finite state machine (FSM) which controls the overall LBIST operation.

The LBIST solution that is supported by LSTC is based on a STUMPS (Self-Test Using MISR and Parallel SRPG) architecture and supports run-time programming. The inserted LBIST logic uses:

- A pseudo-random pattern generator (PRPG), also referred to as Shift Register Pattern Generator (SRPG), to generate input patterns that are applied to the scan channels.
- A multiple input signature register (MISR) to obtain the response to these test input patterns. An incorrect MISR output indicates a defect in the chip.

Once the test is initiated, device will isolate the unit-under-test (UUT) boundary outputs, enter into scan test mode, perform self-test and issue a reset to enter the functional mode at last. Once initiated, the test will run for a pre-determined duration and LSTC hardware will check the signature at the end of this duration and issue a pass/fail status. The key features of LSTC based self-test as listed below:-

- Tests the entire digital logic of unit-under-test (UUT) during start-up and/or at periodic intervals with minimal area addition as compared to stored-pattern based Self-Test Controller (STC).
- Complete flexibility with respect to coverage improvements post-silicon. LBIST IP counters can be reconfigured to achieve given coverage and golden MISR can be calculated post-silicon.
- Provides a sanity check (Failure Insertion) to verify the LSTC functionality.
- LSTC facilitates complete isolation of UUT from the rest of the system during self-test.
- Time-out Monitoring for the self-test run as a fail-safe feature.
- LBIST debugging possible by reading the MISR data of the last pattern for a particular LBIST run.
- Capture power reduction using dead cycles before and after capture pulses.
- Shift power reduction using scan window counter of LBIST IP. The LBIST IP will internally divide the clock based on this counter value and the shift frequency can be reduced.

22.1.1.3 LSTC Block Diagram

The LSTC module is composed of following blocks of logic which is shown in Figure 2.

- LSTC State Machine
- LSTC REGS
- LSTC TIMEOUT
- LSTC SYNCH

The diagram below contains the hierarchical details of LSTC.

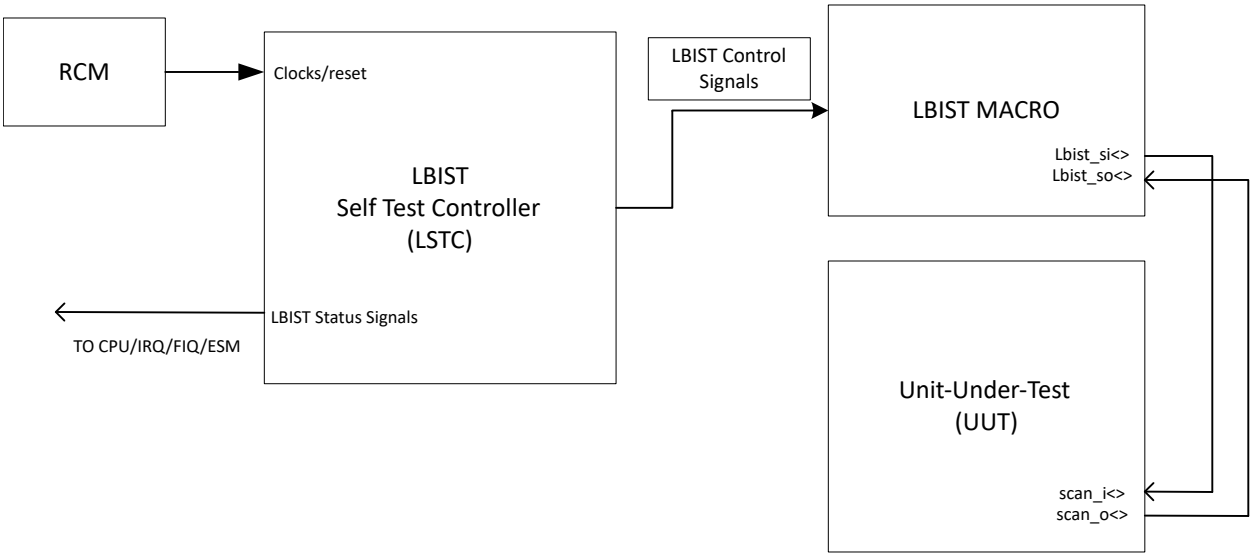


Figure 22-1. LSTC Block Diagram

22.1.1.4 LSTC Flow Diagram

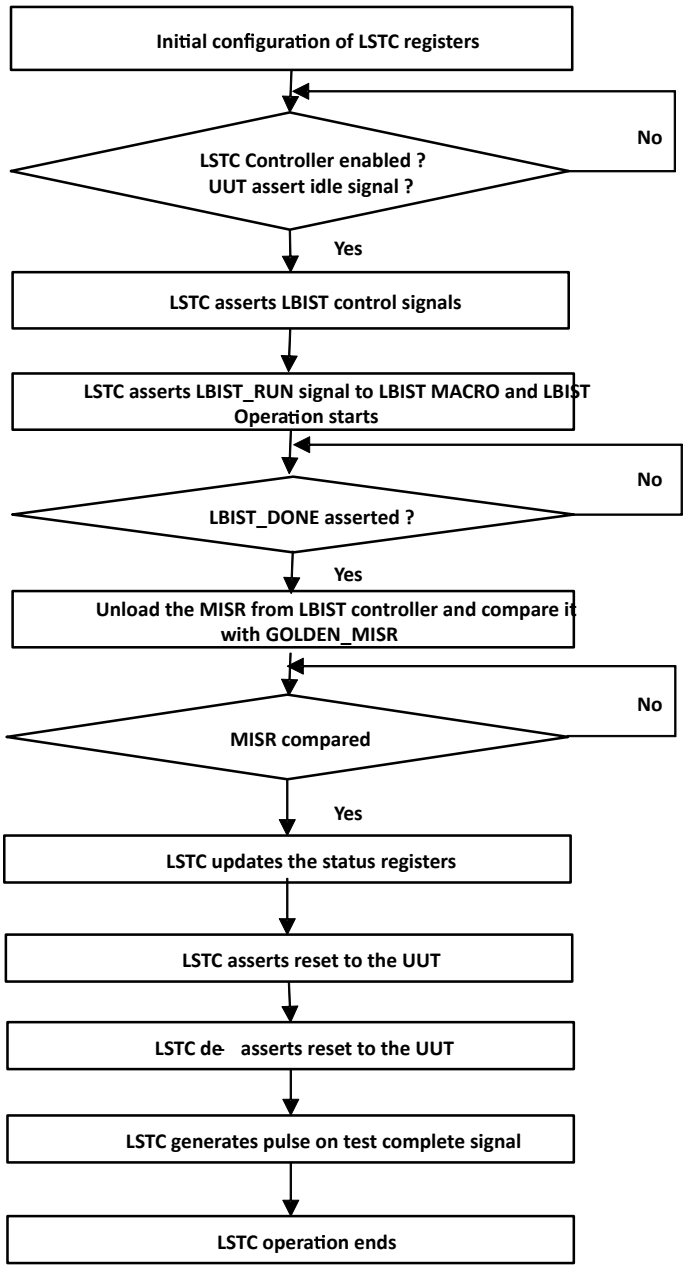


Figure 22-2. LSTC Flow Diagram

22.1.1.5 LSTC State Machine Flow

This section describes the LSTC state machine flow. Once the LSTC is initiated, the state machine traverses through the following states and finally based on the MISR compare, issues a fail pulse.

Table 22-1. LSTC FSM States

S. No.	State Name	State Description	Exit Criteria
1	**LSTC_IDLE**	The initial state of any LSTC run. The POR signal causes the FSM to asynchronously enter this state. The FSM will wait here until LSTC_ENABLE_KEY register is configured (4'b1010) and DUT_IDLE is asserted.	dut_idle == 1 && lbist_key=4'b1010
2	**LSTC_STATUS_CLEAR**	During this state, the status registers (fail/done) will get cleared.	~1 cycle
3	**LSTC_RESET_ASSERT**	In this state, LSTC Macro reset signal **lstc_macro_reset_o** is asserted and LSTC Macro goes into READY state	~1 cycle
4	**LSTC_RESET_ASSERT_WAIT**	Wait State, CFG count value	~4 cycles
5	**LSTC_RESET_DEASSERT**	LSTC Macro comes out of reset by de-asserting LSTC Macro reset signal **lstc_macro_reset_o**	~1 cycle
6	**LSTC_RESET_DEASSERT_WAIT**	Wait State, CFG count value	~4 cycles
7	**LSTC_DUT_ISO**	In this state, **lstc_uut_isolate_o** signal is generated which is used to isolate the DUT output signals	~1 cycle
8	**LSTC_DUT_ISO_WAIT**	Wait State, CFG count value	CFG1 count value
9	**LSTC_CLK_ASSERT**	In this state, **lstc_macro_clk_en** signal is generated which is used to turn-on the LSTC CLK (ICG Control) and **lstc_dut_clk_en_o** to stop the functional clocks	~1 cycle
10	**LSTC_CLK_ASSERT_WAIT**	Wait State, CFG count value	CFG2 count value
11	**LSTC_MUX_ASSERT**	**lstc_set_en** signal generated in this state can be used as a MUX select line for the signals which need to be at specific levels during LSTC run	~1 cycle
12	**LSTC_MUX_ASSERT_WAIT**	Wait State, CFG count value	CFG3 count value
13	**LSTC_LBIST_RUN**	In this state, **lstc_macro_run_o** signal is asserted which will start the LSTC execution. The LSTC Macro traverses through BIST INIT, SCAN TEST, SET TEST, RESET TEST, STATIC TEST, DONE states	~1 cycle
14	**LSTC_LBIST_DONE_WAIT**	Waiting for **lstc_macro_bist_done** signal to go high	LSTC RUN TIME
15	**LSTC_COMP**	Comparing the actual MISR with expected MISR in this state and consecutively asserting **lstc_fail_o** if there is a fail.	~4 cycles

Table 22-1. LSTC FSM States (continued)

S. No.	State Name	State Description	Exit Criteria
16	**LSTC_CLK_DEASSERT**	This state de-asserts **lstc_macro_clk_en_o** going to the LSTC MACRO and asserts **lstc_dut_clk_en_o** to the DUT	~1 cycle
17	**LSTC_CLK_DEASSERT_WAIT**	Wait State, CFG count value	CFG4 count value
18	**LSTC_DUT_RESET_ASSERT**	This state asserts **lstc_uut_reset_o** signal	~1 cycle
19	**LSTC_DUT_RESET_ASSERT_WAIT**	Wait State, CFG count value	CFG5 count value
20	**LSTC_MUX_DEASSERT**	This state de-asserts **lstc_macro_run_o** and **lstc_set_en_o** going to the LSTC MACRO and DUT	~1 cycle
21	**LSTC_MUX_DEASSERT_WAIT**	Wait State, CFG count value	CFG6 count value
22	**LSTC_DUT_ISO_DEASSERT**	This state de-asserts **lstc_uut_isolate_o** signal	~1 cycle
23	**LSTC_DUT_ISO_DEASSERT_WAIT**	Wait State, CFG count value	CFG7 count value
24	**LSTC_DUT_RESET_DEASSERT**	This state de-asserts **lstc_uut_reset_o** signal.	~1 cycle
25	**LSTC_TEST_COMPLETE**	This state generates a pulse **lbist_test_complete** for the CPU.	~4 cycles
26	**LSTC_COMPLETE**	After generation of test_complete, the FSM clears the RESET and KEY register and goes to the IDLE state	~1 cycle

22.1.1.6 LSTC Components Description

22.1.1.6.1 LSTC State Machine

This block generates the control signals and pattern counters signals and sends them to LBIST Macro controller. This block also controls the UUT handshake with respect to UUT Isolation, Reset Assertion/De-assertion and Clocks Assertion/De-assertion. The sequence of operation is defined in the Flow chart under section 1.7. The timing protocol and flow diagrams are covered in Timing Diagrams.

22.1.1.6.2 LSTC 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, how many pattern intervals to be completed before stopping etc. The register block also captures various status information of self test for the user.

22.1.1.6.3 LSTC Time-out Block

This block implements the timeout monitoring logic which will act as a fail-safe feature in LSTC. If the LSTC or LBIST Macro is not able to work properly, time-out monitoring will issue timeout error which can be processed by the processor.

22.1.1.6.4 LSTC Synchronization Block

This block implements the synchronization of control signals between VBUSP clock domain and LSTC clock domain. This block is also responsible for providing ATPG overrides for certain output signals.

22.1.1.7 LSTC State Machine Waveform

The sequencing of control signals to LBIST Macro and UUT during LSTC run as specified in section 1.4 are displayed in below waveforms.

Figure 22-3. LSTC Sequence Waveform Diagram

22.1.1.8 Interface Description

22.1.1.8.1 Generic Parameters

Parameter	Definition
MISR_LENGTH	MISR length Supported values 1 to 256

22.1.1.8.2 Boundary Interface Signals

Table 22-2. Boundary Interface Signals

Signal Name	I/O	I/O Clock Domain	Bus	Description
lstc_uut_idle_i	I	Lstc_clk	-	IDLE indication from the UUT showing readiness for self-test
lstc_uut_idle_override_i	I		-	WFI Override for IDLE indication
lstc_macro_done_i	I		-	LBIST Done indication from LBIST Macro
lstc_macro_actual_misr_i	I		MISR_LENGTH-1	Actual MISR from LBIST Codec for comparison
lstc_macro_static_pc_o	O		13:0	Static patterns count for LBIST run.
lstc_macro_set_pc_o	O		5:0	Set patterns count for LBIST run.
lstc_macro_reset_pc_o	O		5:0	Reset patterns count for LBIST run.
lstc_macro_scan_pc_o	O		5:0	Scan patterns count for LBIST run.
lstc_macro_sw_pc_o	O		3:0	Scan window count for LBIST run
lstc_macro_se_pc_o	O		3:0	Scan enable count for LBIST run
lstc_macro_cap_pc_o	O		3:0	Capture idle count for LBIST run
lstc_macro_resetn_o	O		-	LBIST Macro reset for every LSTC run
lstc_macro_clk_en_o	O		-	Macro clock enable control signal
lstc_macro_run_o	O		-	LBIST Macro run signal
lstc_uut_isolate_o	O		-	UUT isolate signal
lstc_uut_clk_dis_o	O		-	UUT clock disable control signal
lstc_set_en_o	O		-	UUT set enable signal for static multiplexers
lstc_uut_resetn_o	O		-	UUT combined reset signal

22.1.1.8.3 System Interface

Signal Name	I/O	Bus	Description
mod_por_rst_n	I	-	Power up reset from the PLL wrapper
mod_g_rst_n	I	-	System reset from the SYS module.
Vbusp_clk	I	-	Clock for the MMR interface operation, typically VBUSP_CLK.
lstc_clk	I	-	Functional clock connected to UUT and LBIST Macro

22.1.1.8.4 Error Status Module (ESM) and Vectored Interrupt Module (VIM) Interface

Signal Name	I/O	Bus	Description
lstc_error_pulse_o	O	-	ORed signal (VBUSP domain) of lstc_testerr_o and nstc_timeouterr_o.
lstc_test_complete_pulse_o	O	-	Self test complete indication pulse (VBUSP domain) that can be used as an interrupt request. Can be used for non-CPU instances of NSTC as interrupt source to CPU to get the indication self-test run completion.

22.1.1.8.5 PCR Interface

Signal Name	I/O	Bus	Description
vbusp_req_i	I	-	Peripheral select for NSTC register file.
vbusp_dir_i	I	-	Read write indicator - low for write
vbusp_prot_i	I	-	1=privilege mode access.
vbusp_addr_i	I	31:0	VBUSP Address
vbusp_wdata_i	I	31:0	write data bus
vbusp_byten_i	I	3:0	Byte enable signals
vbusp_rready_o	O	-	Read READY returned from slave
vbusp_wready_o	O	-	Write READY returned from slave
vbusp_aerror_o	O	-	VBUSP address error returned from slave. When STC registers are written in non-privilege mode, this error is asserted
vbusp_rdata_o	O	31:0	read data bus

22.1.2 LSTC Registers

Table 22-3 lists the memory-mapped registers for the LSTC registers. All register offset addresses not listed in Table 22-3 should be considered as reserved locations and the register contents should not be modified.

Table 22-3. LSTC Registers

Offset	Acronym	Register Name	Section
0h	PID	PID register	Go
4h	RESERVED0		Go
8h	LSTC_ENABLE		Go
Ch	LSTC_STATUS		Go
10h	LSTC_CFG_CNTRS		Go
14h	LSTC_LBIST_MACRO_CNTRS		Go
18h	LSTC_TO_PRELOAD		Go
1Ch	LSTC_MACRO_SC		Go
20h	LSTC_GOLDEN_MISR_0		Go
24h	LSTC_GOLDEN_MISR_1		Go
28h	LSTC_GOLDEN_MISR_2		Go
2Ch	LSTC_GOLDEN_MISR_3		Go
30h	LSTC_GOLDEN_MISR_4		Go
34h	LSTC_GOLDEN_MISR_5		Go
38h	LSTC_GOLDEN_MISR_6		Go
3Ch	LSTC_GOLDEN_MISR_7		Go
40h	LSTC_ACTUAL_MISR_0		Go
44h	LSTC_ACTUAL_MISR_1		Go
48h	LSTC_ACTUAL_MISR_2		Go
4Ch	LSTC_ACTUAL_MISR_3		Go
50h	LSTC_ACTUAL_MISR_4		Go
54h	LSTC_ACTUAL_MISR_5		Go
58h	LSTC_ACTUAL_MISR_6		Go
5Ch	LSTC_ACTUAL_MISR_7		Go
1008h	LOCK0_KICK0	- KICK0 component	Go
100Ch	LOCK0_KICK1	- KICK1 component	Go
1010h	intr_raw_status	Interrupt Raw Status/Set Register	Go
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	Go
1018h	intr_enable	Interrupt Enable register	Go
101Ch	intr_enable_clear	Interrupt Enable Clear register	Go
1020h	eoi	EOI register	Go
1024h	fault_address	Fault Address register	Go
1028h	fault_type_status	Fault Type Status register	Go
102Ch	fault_attr_status	Fault Attribute Status register	Go
1030h	fault_clear	Fault Clear register	Go

Complex bit access types are encoded to fit into small table cells. Table 22-4 shows the codes that are used for access types in this section.

Table 22-4. LSTC Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		

Table 22-4. LSTC Access Type Codes (continued)

Access Type	Code	Description
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

22.1.2.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in Table 22-5.

Return to the Summary Table.

PID register

Table 22-5. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

22.1.2.2 RESERVED0 Register (Offset = 4h) [Reset = X]

RESERVED0 is shown in Table 22-6.

Return to the Summary Table.

Table 22-6. RESERVED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	wphres	R/W	0h	Reserved
23-16	RESERVED	R/W	X	
15-8	rores	R	0h	Reserved
7-0	rwres	R/W	0h	Reserved

22.1.2.3 LSTC_ENABLE Register (Offset = 8h) [Reset = X]

LSTC_ENABLE is shown in Table 22-7.

Return to the Summary Table.

Table 22-7. LSTC_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	X	
4	LSTC_UUT_IDLE_OVR	R/W	0h	
3-0	LSTC_KEY_ENA	R/W	0h	

22.1.2.4 LSTC_STATUS Register (Offset = Ch) [Reset = X]

LSTC_STATUS is shown in Table 22-8.

Return to the Summary Table.

Table 22-8. LSTC_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	X	
12-8	LSTC_CURRENT_STATUS	R	0h	
7	LSTC_CORE_ERROR	R	0h	
6	LSTC_TO_ERROR	R	0h	
5	LSTC_FAIL	R	0h	
4	LSTC_DONE	R	0h	
3-0	LSTC_ACTIVE	R	0h	

22.1.2.5 LSTC_CFG_CNTRS Register (Offset = 10h) [Reset = X]

LSTC_CFG_CNTRS is shown in Table 22-9.

Return to the Summary Table.

Table 22-9. LSTC_CFG_CNTRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27-24	LSTC_SM_CFG_CNTR7	R/W	8h	
23-20	LSTC_SM_CFG_CNTR6	R/W	8h	
19-16	LSTC_SM_CFG_CNTR5	R/W	8h	
15-12	LSTC_SM_CFG_CNTR4	R/W	8h	
11-8	LSTC_SM_CFG_CNTR3	R/W	8h	
7-4	LSTC_SM_CFG_CNTR2	R/W	8h	
3-0	LSTC_SM_CFG_CNTR1	R/W	8h	

22.1.2.6 LSTC_LBIST_MACRO_CNTRS Register (Offset = 14h) [Reset = 00010820h]

LSTC_LBIST_MACRO_CNTRS is shown in Table 22-10.

Return to the Summary Table.

Table 22-10. LSTC_LBIST_MACRO_CNTRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	STATIC_PC_DEF	R/W	0h	
17-12	LSTC_SCAN_PC_DEF	R/W	10h	
11-6	LSTC_RESET_PC_DEF	R/W	20h	
5-0	LSTC_SET_PC_DEF	R/W	20h	

22.1.2.7 LSTC_TO_PRELOAD Register (Offset = 18h) [Reset = FFFFFFFFh]

LSTC_TO_PRELOAD is shown in Table 22-11.

Return to the Summary Table.

Table 22-11. LSTC_TO_PRELOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_TO_PRELOAD	R/W	FFFFFFFFh	

22.1.2.8 LSTC_MACRO_SC Register (Offset = 1Ch) [Reset = X]

LSTC_MACRO_SC is shown in Table 22-12.

Return to the Summary Table.

Table 22-12. LSTC_MACRO_SC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	X	
11-8	LSTC_MACRO_CAP_DELAY	R/W	4h	
7-4	LSTC_MACRO_SE_DELAY	R/W	2h	
3-0	LSTC_MACRO_SW_DELAY	R/W	1h	

22.1.2.9 LSTC_GOLDEN_MISR_0 Register (Offset = 20h) [Reset = 00000000h]

LSTC_GOLDEN_MISR_0 is shown in Table 22-13.

Return to the Summary Table.

Table 22-13. LSTC_GOLDEN_MISR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_0	R/W	0h	

22.1.2.10 LSTC_GOLDEN_MISR_1 Register (Offset = 24h) [Reset = 00000000h]

LSTC_GOLDEN_MISR_1 is shown in Table 22-14.

Return to the Summary Table.

Table 22-14. LSTC_GOLDEN_MISR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_1	R/W	0h	

22.1.2.11 LSTC_GOLDEN_MISR_2 Register (Offset = 28h) [Reset = 00000000h]

LSTC_GOLDEN_MISR_2 is shown in Table 22-15.

Return to the Summary Table.

Table 22-15. LSTC_GOLDEN_MISR_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_2	R/W	0h	

22.1.2.12 LSTC_GOLDEN_MISR_3 Register (Offset = 2Ch) [Reset = 00000000h]

LSTC_GOLDEN_MISR_3 is shown in Table 22-16.

Return to the Summary Table.

Table 22-16. LSTC_GOLDEN_MISR_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_3	R/W	0h	

22.1.2.13 LSTC_GOLDEN_MISR_4 Register (Offset = 30h) [Reset = 00000000h]

LSTC_GOLDEN_MISR_4 is shown in Table 22-17.

Return to the Summary Table.

Table 22-17. LSTC_GOLDEN_MISR_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_4	R/W	0h	

22.1.2.14 LSTC_GOLDEN_MISR_5 Register (Offset = 34h) [Reset = 00000000h]

LSTC_GOLDEN_MISR_5 is shown in Table 22-18.

Return to the Summary Table.

Table 22-18. LSTC_GOLDEN_MISR_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_5	R/W	0h	

22.1.2.15 LSTC_GOLDEN_MISR_6 Register (Offset = 38h) [Reset = 00000000h]

LSTC_GOLDEN_MISR_6 is shown in Table 22-19.

Return to the Summary Table.

Table 22-19. LSTC_GOLDEN_MISR_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_6	R/W	0h	

22.1.2.16 LSTC_GOLDEN_MISR_7 Register (Offset = 3Ch) [Reset = 00000000h]

LSTC_GOLDEN_MISR_7 is shown in Table 22-20.

Return to the Summary Table.

Table 22-20. LSTC_GOLDEN_MISR_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_GOLDEN_MISR_7	R/W	0h	

22.1.2.17 LSTC_ACTUAL_MISR_0 Register (Offset = 40h) [Reset = 00000000h]

LSTC_ACTUAL_MISR_0 is shown in Table 22-21.

Return to the Summary Table.

Table 22-21. LSTC_ACTUAL_MISR_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_0	R	0h	

22.1.2.18 LSTC_ACTUAL_MISR_1 Register (Offset = 44h) [Reset = 00000000h]

LSTC_ACTUAL_MISR_1 is shown in Table 22-22.

Return to the Summary Table.

Table 22-22. LSTC_ACTUAL_MISR_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_1	R	0h	

22.1.2.19 LSTC_ACTUAL_MISR_2 Register (Offset = 48h) [Reset = 00000000h]

LSTC_ACTUAL_MISR_2 is shown in Table 22-23.

Return to the Summary Table.

Table 22-23. LSTC_ACTUAL_MISR_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_2	R	0h	

22.1.2.20 LSTC_ACTUAL_MISR_3 Register (Offset = 4Ch) [Reset = 00000000h]

LSTC_ACTUAL_MISR_3 is shown in Table 22-24.

Return to the Summary Table.

Table 22-24. LSTC_ACTUAL_MISR_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_3	R	0h	

22.1.2.21 LSTC_ACTUAL_MISR_4 Register (Offset = 50h) [Reset = 00000000h]

LSTC_ACTUAL_MISR_4 is shown in Table 22-25.

Return to the Summary Table.

Table 22-25. LSTC_ACTUAL_MISR_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_4	R	0h	

22.1.2.22 LSTC_ACTUAL_MISR_5 Register (Offset = 54h) [Reset = 00000000h]

LSTC_ACTUAL_MISR_5 is shown in Table 22-26.

Return to the Summary Table.

Table 22-26. LSTC_ACTUAL_MISR_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_5	R	0h	

22.1.2.23 LSTC_ACTUAL_MISR_6 Register (Offset = 58h) [Reset = 00000000h]

LSTC_ACTUAL_MISR_6 is shown in Table 22-27.

Return to the Summary Table.

Table 22-27. LSTC_ACTUAL_MISR_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_6	R	0h	

22.1.2.24 LSTC_ACTUAL_MISR_7 Register (Offset = 5Ch) [Reset = 00000000h]

LSTC_ACTUAL_MISR_7 is shown in Table 22-28.

Return to the Summary Table.

Table 22-28. LSTC_ACTUAL_MISR_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LSTC_ACTUAL_MISR_7	R	0h	

22.1.2.25 LOCK0_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0_KICK0 is shown in Table 22-29.

Return to the Summary Table.

- KICK0 component

Table 22-29. LOCK0_KICK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

22.1.2.26 LOCK0_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0_KICK1 is shown in Table 22-30.

Return to the Summary Table.

- KICK1 component

Table 22-30. LOCK0_KICK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

22.1.2.27 intr_raw_status Register (Offset = 1010h) [Reset = X]

intr_raw_status is shown in Table 22-31.

Return to the Summary Table.

Interrupt Raw Status/Set Register

Table 22-31. intr_raw_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

22.1.2.28 intr_enabled_status_clear Register (Offset = 1014h) [Reset = X]

intr_enabled_status_clear is shown in Table 22-32.

Return to the Summary Table.

Interrupt Enabled Status/Clear register

Table 22-32. intr_enabled_status_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

22.1.2.29 intr_enable Register (Offset = 1018h) [Reset = X]

intr_enable is shown in Table 22-33.

Return to the Summary Table.

Interrupt Enable register

Table 22-33. intr_enable Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

22.1.2.30 intr_enable_clear Register (Offset = 101Ch) [Reset = X]

intr_enable_clear is shown in Table 22-34.

Return to the Summary Table.

Interrupt Enable Clear register

Table 22-34. intr_enable_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

ADVANCE INFORMATION

22.1.2.31 eoi Register (Offset = 1020h) [Reset = X]

eoi is shown in Table 22-35.

Return to the Summary Table.

EOI register

Table 22-35. eoi Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

22.1.2.32 fault_address Register (Offset = 1024h) [Reset = 00000000h]

fault_address is shown in Table 22-36.

Return to the Summary Table.

Fault Address register

Table 22-36. fault_address Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

22.1.2.33 fault_type_status Register (Offset = 1028h) [Reset = X]

fault_type_status is shown in Table 22-37.

Return to the Summary Table.

Fault Type Status register

Table 22-37. fault_type_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	X	
6	fault_ns	R	0h	Non-secure access.

Table 22-37. fault_type_status Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	fault_type	R	0h	Fault Type 10_ 0000 = Supervisor read fault - priv = 1 dir = 1 dtype != 1 01_ 0000 = Supervisor write fault - priv = 1 dir = 0 00_ 1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_ 0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_ 0010 = User write fault - priv = 0 dir = 0 00_ 0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_ 0000 = No fault

22.1.2.34 fault_attr_status Register (Offset = 102Ch) [Reset = 00000000h]

fault_attr_status is shown in Table 22-38.

Return to the Summary Table.

Fault Attribute Status register

Table 22-38. fault_attr_status Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

22.1.2.35 fault_clear Register (Offset = 1030h) [Reset = X]

fault_clear is shown in Table 22-39.

Return to the Summary Table.

Fault Clear register

Table 22-39. fault_clear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	X	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

22.1.3 LSTC Sequences

22.1.3.1 Basic Run Sequence

The following tables capture the details of the Basic LSTC run:

Table 22-40. Basic LSTC Sequence

No.	Steps	Register/Bit Field/Programming	Value
1	Configure the number of patterns to be executed.	LSTC.LSTC_MACRO_PC.LSTC_SET_PC_DEF LSTC.LSTC_MACRO_PC.LSTC_RESET_PC_DEF LSTC.LSTC_MACRO_PC.LSTC_SCAN_PC_DEF LSTC.LSTC_MACRO_PC.LSTC_STATIC_PC_DEF	TBD
2	Scan mode configuration. Configure the scan window counter.	LSTC.LSTC_MACRO_SC.LSTC_MACRO_SW_DELAY	0x01

Table 22-40. Basic LSTC Sequence (continued)

No.	Steps	Register/Bit Field/Programming	Value
3	Scan mode configuration. Configure the capture idle counter.	LSTC.LSTC_MACRO_SC.LSTC_MACRO_CAP_DELAY	0x04
4	Scan mode configuration. Configure the scan enable delay counter.	LSTC.LSTC_MACRO_SC.LSTC_MACRO_SE_DELAY	0x1
5	Program the Timer Register for max run time	LSTC.LSTC_TO_PRELOAD	
6	Configure the Golden MISR.	LSTC.LSTC_GOLDEN_MISR_0 LSTC.LSTC_GOLDEN_MISR_0 .. .. .. LSTC.LSTC_GOLDEN_MISR_7	TBD
10	Kick off the test	LSTC.LSTC_ENABLE.LSTC_KEY_ENA	0x5
11	Wait for Test done Interrupt or ESM error	LSTC.LSTC_STATUS.LSTC_DONE	0x1
12	Read the status register to check the STC test completion.	LSTC.LSTC_STATUS.LSTC_FAIL	0x1(READ)
			(READ) 0x0 - No failure

22.1.3.2 Failure Insertion Run Sequence

The following tables capture the details of the Failure Insertion LSTC run.

Table 22-41. Fail Insertion LSTC Sequence

No.	Steps	Register/Bit Field/Programming	Value
1	Configure the number of patterns to be executed.	LSTC.LSTC_MACRO_PC.LSTC_SET_PC_DEF LSTC.LSTC_MACRO_PC.LSTC_RESET_PC_DEF LSTC.LSTC_MACRO_PC.LSTC_SCAN_PC_DEF LSTC.LSTC_MACRO_PC.LSTC_STATIC_PC_DEF	TBD
2	Scan mode configuration. Configure the scan window counter.	LSTC.LSTC_MACRO_SC.LSTC_MACRO_SW_DELAY	0x01
3	Scan mode configuration. Configure the capture idle counter.	LSTC.LSTC_MACRO_SC.LSTC_MACRO_CAP_DELAY	0x04
4	Scan mode configuration. Configure the scan enable delay counter.	LSTC.LSTC_MACRO_SC.LSTC_MACRO_SE_DELAY	0x1
5	Program the Timer Register for max run time	LSTC.LSTC_TO_PRELOAD	
6	Configure the Golden MISR.	LSTC.LSTC_GOLDEN_MISR_0 LSTC.LSTC_GOLDEN_MISR_0 .. .. .. LSTC.LSTC_GOLDEN_MISR_7	Wrong Golden MISR for LSTC to fail.

Table 22-41. Fail Insertion LSTC Sequence (continued)

No.	Steps	Register/Bit Field/Programming	Value
10	Kick off the test	LSTC.LSTC_ENABLE.LSTC_KEY_ENA	0x5
11	Wait for Test done Interrupt or ESM error	LSTC.LSTC_STATUS.LSTC_DONE	0x1
12	Read the status register to check the STC test completion.	LSTC.LSTC_STATUS.LSTC_FAIL	0x1(READ)
			(READ) 0x0 - No failure

22.1.3.3 Timing Waveforms

Figure 22-4. LSTC Basic Run

Figure 22-5. LSTC Error Run

Figure 22-6. LSTC Back to Back Run

Figure 22-7. LSTC Time-out Run

22.2 Dual Clock Comparator (DCC)

This section describes the dual-clock comparator (DCC) module.

22.2.1 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. Specifically, DCC is designed to detect drifts from the expected clock frequency. This capability can be used to ensure the correct frequency range for several different device clock sources, thereby enhancing the system safety metrics.

22.2.1.1 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

22.2.1.2 Block Diagram

Figure 22-8 illustrates the main concept of the DCC module.

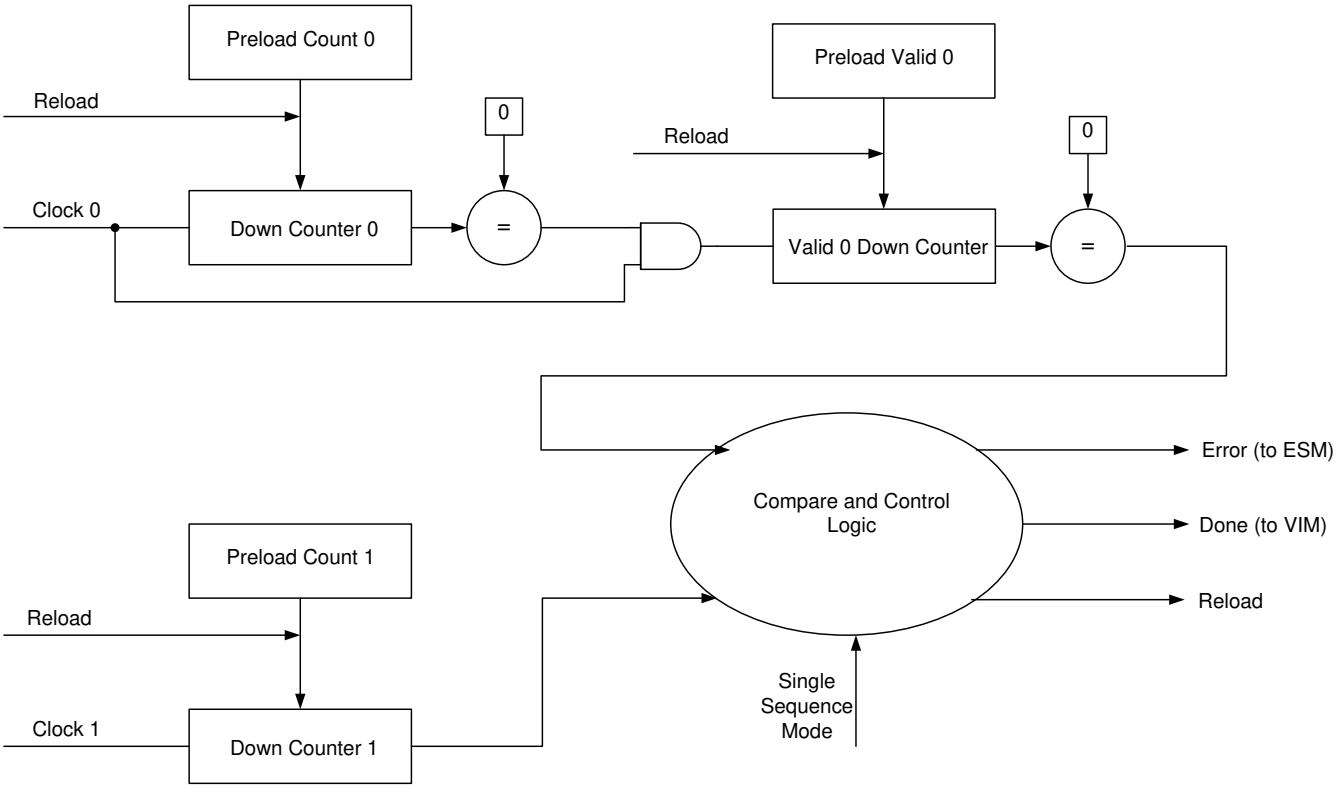


Figure 22-8. Block Diagram

22.2.2 Module Operation

As shown in Figure 22-8, 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:

22.2.2.1 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.

22.2.2.1.1 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

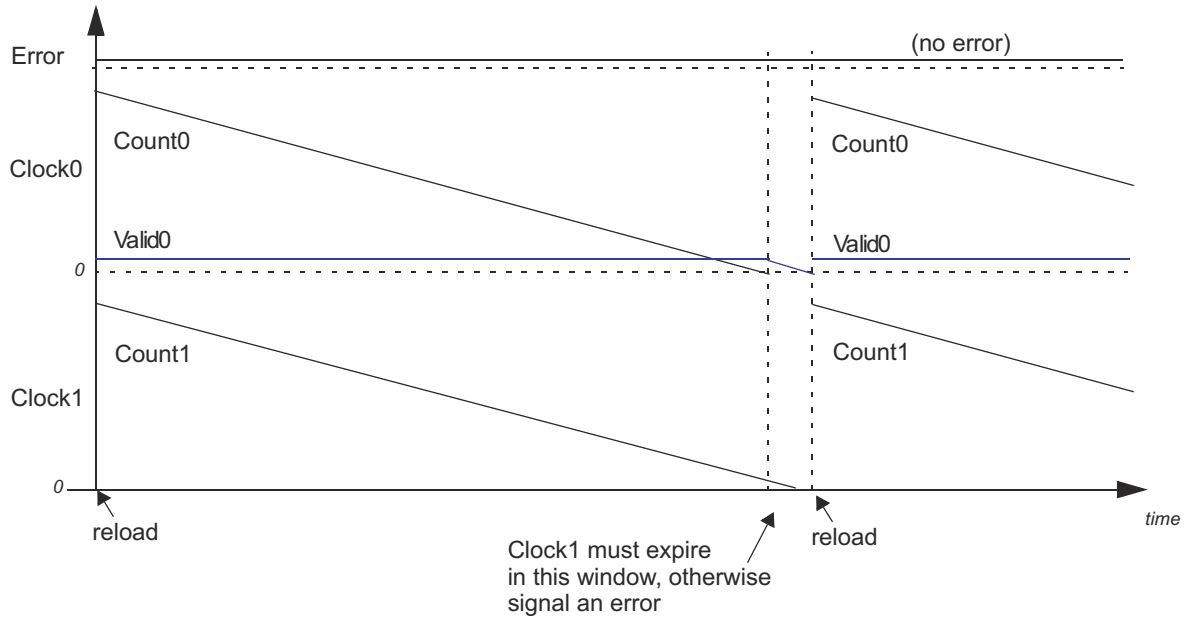


Figure 22-9. Counter Relationship

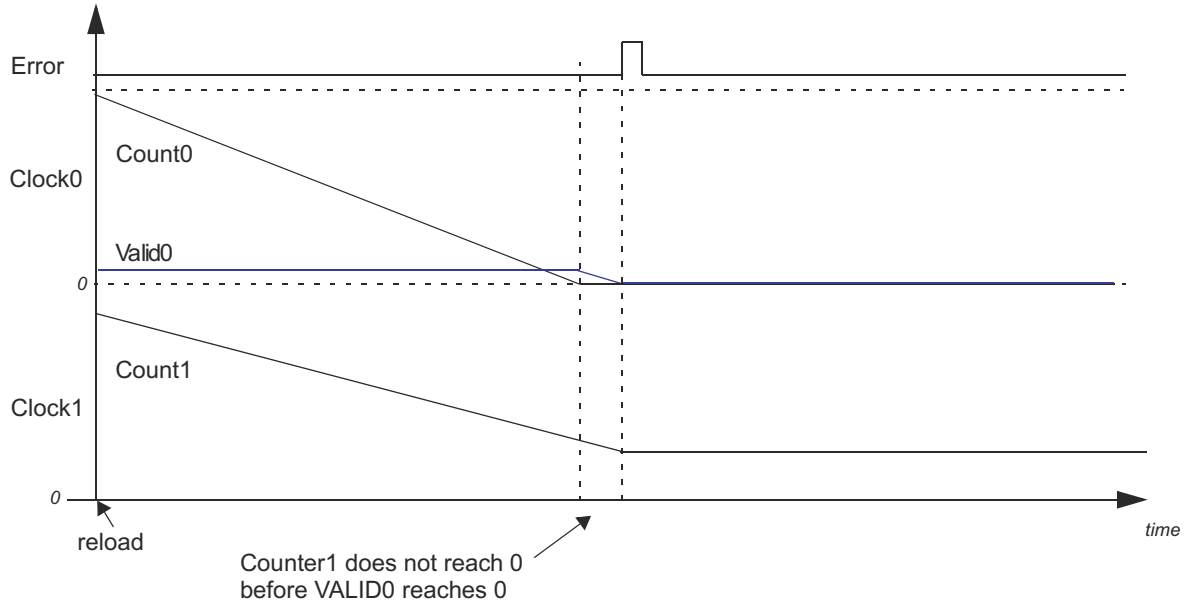


Figure 22-10. Clock1 Slower Than Clock0 - Results in an Error and Stops Counting

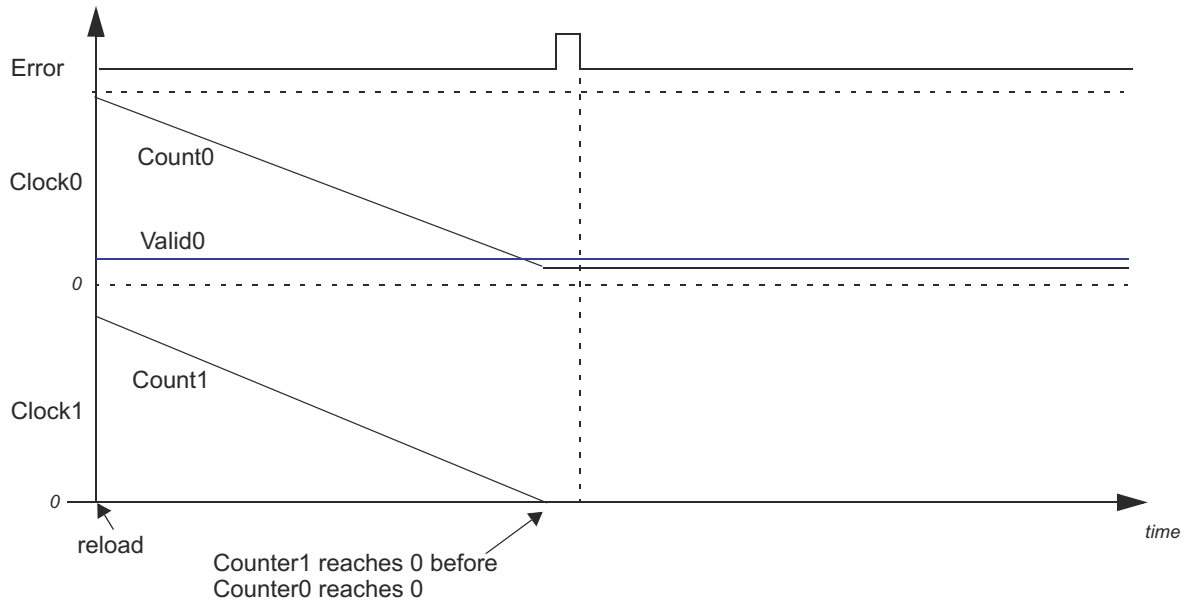


Figure 22-11. Clock1 Faster Than Clock0 - Results in an Error and Stops Counting

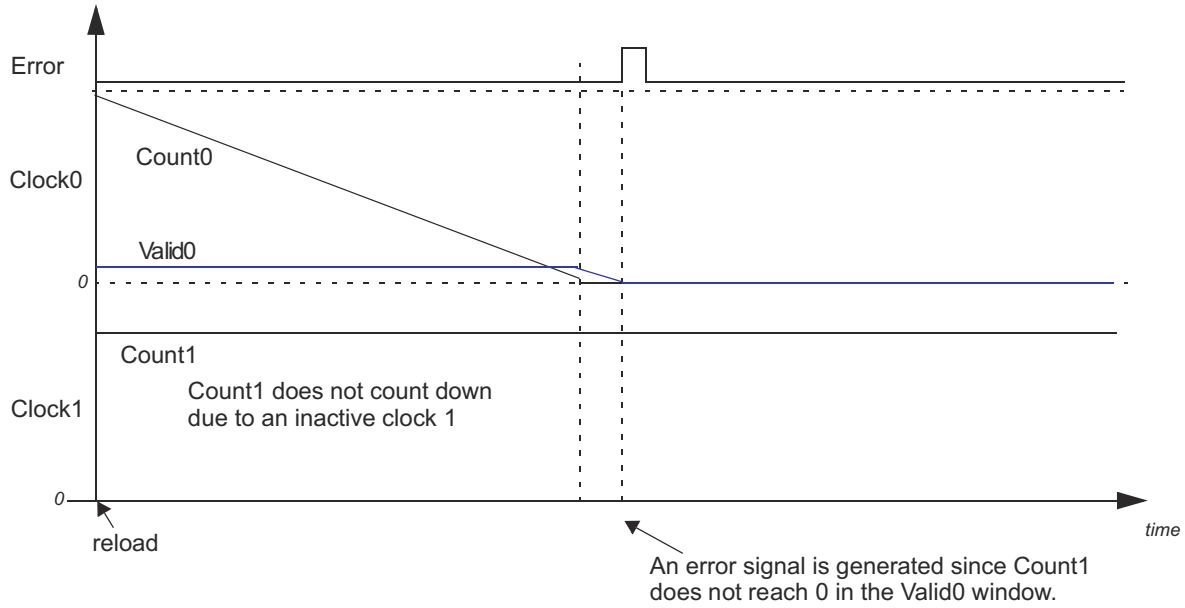


Figure 22-12. Clock1 Not Present - Results in an Error and Stops Counting

ADVANCE INFORMATION

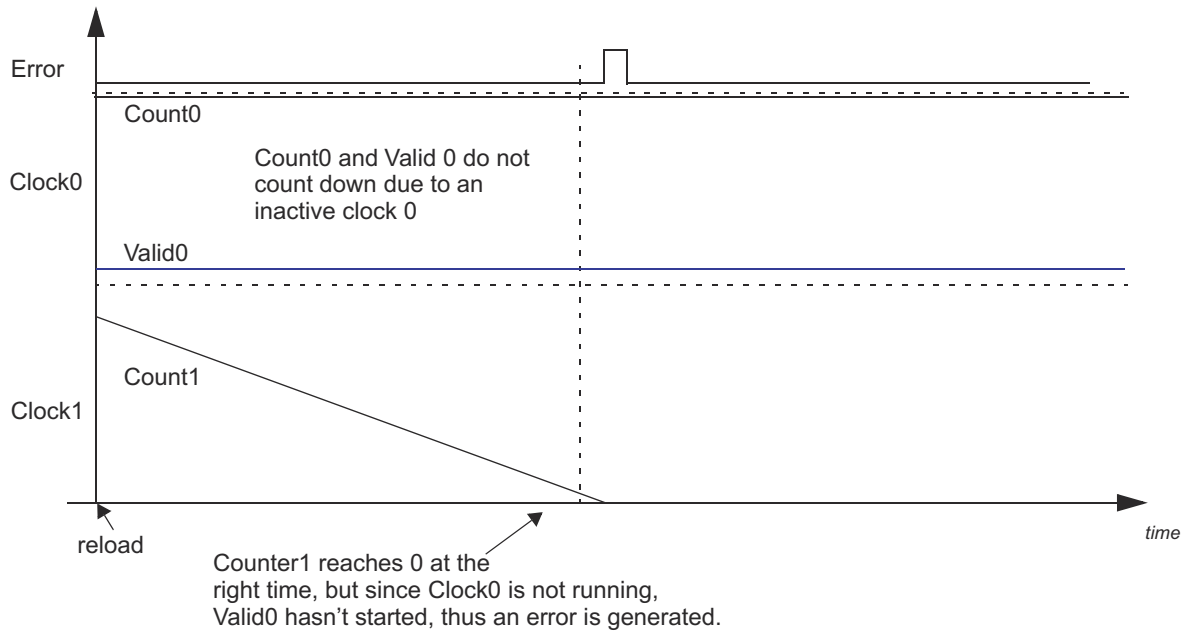


Figure 22-13. Clock0 Not Present - Results in an Error and Stops Counting

22.2.2.2 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 10 MHz and the intended duration of the measurement is 500 µs. 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 500-µs measurement period. This means that the average frequency of the HF LPO over this 500-µs period was 4000 cycles / 500 µs, or 8 MHz.

- 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.

22.2.3 APP DCC Integration

1-DCC module has been instantiated in the SOC as part of APP. Clocks to the module are mentioned in the following sections.

22.2.3.1 Input Clock sources

Table 22-42. APP_DCC CLKSRC0 Mapping

Clock Signal	Description	Source	REG Value for Selection
Input0_clksrc[0]	Primary Oscillator Clock	OSC_CLK / XTAL_CLK	0x0
Input0_clksrc[1]	Digital Phase-Locked Loop Clock	PLL_DIG Clock	0x1
Input0_clksrc[2]	Analog Phase–Locked Loop Clock	APLL clock	0x2
Input0_clksrc[3]	RC Internal Oscillator Clock	RCOSC	0x3
Input0_clksrc[4]	Counter 0 Clock Source	RTL_CLK_IN	0x4

Table 22-43. APP_DCC CLKSRC1 Mapping

Clock Signal	Description	Source	REG Value for Selection
Input1_clksrc[0]	Analog Phase–Locked Loop Clock or Digital Phase-Locked Loop Clock	fast_clk(muxed apll and pll_dig_clk)(root mux)	0x0
Input1_clksrc[1]	Front End Control Clock Module	FECSS_GCM	0x1
Input1_clksrc[2]	General Purpose ADC Clock	GPADC CLK	0x2
Input1_clksrc[3]	Digital Front End clock	RAMPGEN/DFE CLK	0x3
Input1_clksrc[4]	Divided down version of the synthesizer clock	SYNTH_CLK	0x4

Table 22-43. APP_DCC CLKSRC1 Mapping (continued)

Clock Signal	Description	Source	REG Value for Selection
Input1_clksrc[5]	Multiplier Delay Locked Loop clock	MDLL_CLK	0x5
Input1_clksrc[6]	LIN Clock	LIN_CLK	0x6
Input1_clksrc[7]	Primary Oscillator Clock	OSC_CLK / XTAL	0x7
Input1_clksrc[8]	Application Sub-System Clock	APPSS_GCM	0x8
Input1_clksrc[9]	CANBUS controller clock	CANFD_GCM	0x9
Input1_clksrc[10]	RC Internal Oscillator	RCOSC	0x10
Input1_clksrc[11]	Counter 1 Clock Source	RTL_CLK_IN	0x11

22.2.4 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.

22.2.5 APP_DCC Registers

Table 22-44 lists the memory-mapped registers for the APP_DCC registers. All register offset addresses not listed in Table 22-44 should be considered as reserved locations and the register contents should not be modified.

Table 22-44. APP_DCC Registers

Offset	Acronym	Register Name	Section
0h	DCCGCTRL	DCCGCTRL	Go
4h	DCCREV	DCCREV	Go
8h	DCCCNTSEED0	DCCCNTSEED0	Go
Ch	DCCVALIDSEED0	DCCVALIDSEED0	Go
10h	DCCCNTSEED1	DCCCNTSEED1	Go
14h	DCCSTAT	DCCSTAT	Go
18h	DCCCNT0	DCCCNT0	Go
1Ch	DCCVALID0	DCCVALID0	Go
20h	DCCCNT1	DCCCNT1	Go
24h	DCCCLKSSRC1	DCCCLKSSRC1	Go
28h	DCCCLKSSRC0	DCCCLKSSRC0	Go
30h	DCCGCTRL2	DCCGCTRL2	Go
34h	DCCSTATUS2	DCCSTATUS2	Go
38h	DCCERRCNT	DCCERRCNT	Go

Complex bit access types are encoded to fit into small table cells. Table 22-45 shows the codes that are used for access types in this section.

Table 22-45. APP_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

22.2.5.1 DCCGCTRL Register (Offset = 0h) [Reset = 00005555h]

DCCGCTRL is shown in Table 22-46.

Return to the Summary Table.

Starts / stops the counters clears the error signal

Table 22-46. 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

Table 22-46. DCCGCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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

22.2.5.2 DCCREV Register (Offset = 4h) [Reset = 40010300h]

DCCREV is shown in Table 22-47.

Return to the Summary Table.

Module version

Table 22-47. DCCREV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	SCHEME. - (RO)
29-28	NU1	R	0h	Reserved
27-16	FUNC	R	1h	Functional release number - (RO)
15-11	RTL	R	0h	Design Release Number - (RO)
10-8	MAJOR	R	3h	Major Revision Number - (RO)
7-6	CUSTOM	R	0h	Indicates a special version of the module. May not be supported by standard software - (RO)
5-0	MINOR	R	0h	Minor revision number. - (RO)

22.2.5.3 DCCCNTSEED0 Register (Offset = 8h) [Reset = 00000000h]

DCCCNTSEED0 is shown in Table 22-48.

Return to the Summary Table.

Seed value for the counter attached to clock source 0

Table 22-48. 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)

22.2.5.4 DCCVALIDSEED0 Register (Offset = Ch) [Reset = 00000000h]

DCCVALIDSEED0 is shown in Table 22-49.

Return to the Summary Table.

Seed value for the timeout counter attached to clock source 0

Table 22-49. 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

22.2.5.5 DCCCNTSEED1 Register (Offset = 10h) [Reset = 00000000h]

DCCCNTSEED1 is shown in Table 22-50.

Return to the Summary Table.

Seed value for the counter attached to clock source 1

Table 22-50. 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

22.2.5.6 DCCSTAT Register (Offset = 14h) [Reset = 00000000h]

DCCSTAT is shown in Table 22-51.

Return to the Summary Table.

Contains the error & done flag bit

Table 22-51. 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.

22.2.5.7 DCCCNT0 Register (Offset = 18h) [Reset = 00000000h]

DCCCNT0 is shown in Table 22-52.

Return to the Summary Table.

Value of the counter attached to clock source 0

Table 22-52. 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)

22.2.5.8 DCCVALID0 Register (Offset = 1Ch) [Reset = 00000000h]

DCCVALID0 is shown in Table 22-53.

Return to the Summary Table.

Value of the valid counter attached to clock source 0

Table 22-53. 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)

22.2.5.9 DCCCNT1 Register (Offset = 20h) [Reset = 00000000h]

DCCCNT1 is shown in Table 22-54.

Return to the Summary Table.

Value of the counter attached to clock source 1

Table 22-54. 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)

22.2.5.10 DCCCLKSSRC1 Register (Offset = 24h) [Reset = 00000000h]

DCCCLKSSRC1 is shown in Table 22-55.

Return to the Summary Table.

Clock source1 selection control

Table 22-55. DCCCLKSSRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU11	R	0h	Reserved
15-12	KEY_B4	R/W	0h	Key Programing (1010 is the KEY Value)
11-4	NU10	R	0h	Reserved
3-0	CLK_SRC1	R/W	0h	RCOSC Input1_clksrc[10] CANFD_GCM Input1_clksrc[9] APPSS_GCM Input1_clksrc[8] OSC_CLK Input1_clksrc[7] LIN_CLK Input1_clksrc[6] MDLL_CLK Input1_clksrc[5] SYNTH_CLK Input1_clksrc[4] RAMPGEN/DFE CLK Input1_clksrc[3] GPADC CLK Input1_clksrc[2] FECSS_GCM Input1_clksrc[1] fast_clk(muxed apll and pll_dig_clk)(root mux) Input1_clksrc[0]

22.2.5.11 DCCCLKSSRC0 Register (Offset = 28h) [Reset = 00000000h]

DCCCLKSSRC0 is shown in Table 22-56.

Return to the Summary Table.

Clock source0 selection control

Table 22-56. DCCCLKSSRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	NU13	R	0h	Reserved
15-12	KEY_B4	R/W	0h	Key Programing (1010 is the KEY Value)
11-4	NU12	R	0h	Reserved
3-0	CLK_SRC0	R/W	0h	APLL clock 400MHz Input0_clksrc[2] PLL_DIG clock 400MHz Input0_clksrc[1] OSC_CLK Input0_clksrc[0]

22.2.5.12 DCCGCTRL2 Register (Offset = 30h) [Reset = 00000555h]

DCCGCTRL2 is shown in Table 22-57.

Return to the Summary Table.

Global control register 2

Table 22-57. DCCGCTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	NU13	R	0h	
11-8	FIFO_NONERR	R/W	5h	FIFO update on Non-Error Enables/disables FIFO writes without the error event on completion of comparison window. 0101: Counter values are captured to non-full FIFO only upon Error event Others: Write counter values to non-full FIFO upon completion of comparison window regardless of error or not. Note this setting is applicable only in Continuous mode in single shot mode FIFO captures counts only on Error. Note: The user should write 1010 to these enable fields to enable each feature to avoid single soft errors.
7-4	FIFO_READ	R/W	5h	FIFO Read Enable Enables the counter read registers reflect FIFO output instead of live counter value. 0101: Counter value is read directly. Others: FIFO output is read Note: The user should write 1010 to these enable fields to enable each feature to avoid single soft errors.
3-0	CONT_ON_ERR	R/W	5h	Continue on Error enable Continues to next window of comparison despite the error condition. 0101: Comparison and counter reload is stopped from advancing if error is detected. Others: Counters get reloaded with seed and continue counting despite the error condition. Note: The user should write 1010 to these enable fields to enable each feature to avoid single soft errors.

22.2.5.13 DCCSTATUS2 Register (Offset = 34h) [Reset = 00000007h]

DCCSTATUS2 is shown in Table 22-58.

Return to the Summary Table.

FIFO status register

Table 22-58. DCCSTATUS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	NU14	R	0h	Reserved
5	COUNT1_FIFO_FULL	R	0h	Count1 FIFO Full Indicates whether Count1 FIFO is Full. 0: Count1 FIFO is not Full 1: Count1 FIFO is Full.
4	VALID0_FIFO_FULL	R	0h	Valid0 FIFO Full Indicates whether Valid0 FIFO is Full. 0: Valid0 FIFO is not Full 1: Valid0 FIFO is Full.
3	COUNT0_FIFO_FULL	R	0h	Count0 FIFO Full Indicates whether Count0 FIFO is Full. 0: Count0 FIFO is not Full 1: Count0 FIFO is Full.
2	COUNT1_FIFO_EMPTY	R	1h	Count1 FIFO Empty Indicates whether Count1 FIFO is Empty. 0: Count1 FIFO is not empty 1: Count1 FIFO is empty.
1	VALID0_FIFO_EMPTY	R	1h	Valid0 FIFO Empty Indicates whether Valid0 FIFO is Empty. 0: Valid0 FIFO is not empty 1: Valid0 FIFO is empty.
0	COUNT0_FIFO_EMPTY	R	1h	Count0 FIFO Empty Indicates whether Count0 FIFO is Empty. 0: Count0 FIFO is not empty 1: Count0 FIFO is empty.

22.2.5.14 DCCERRCNT Register (Offset = 38h) [Reset = 00000000h]

DCCERRCNT is shown in Table 22-59.

Return to the Summary Table.

Error count register

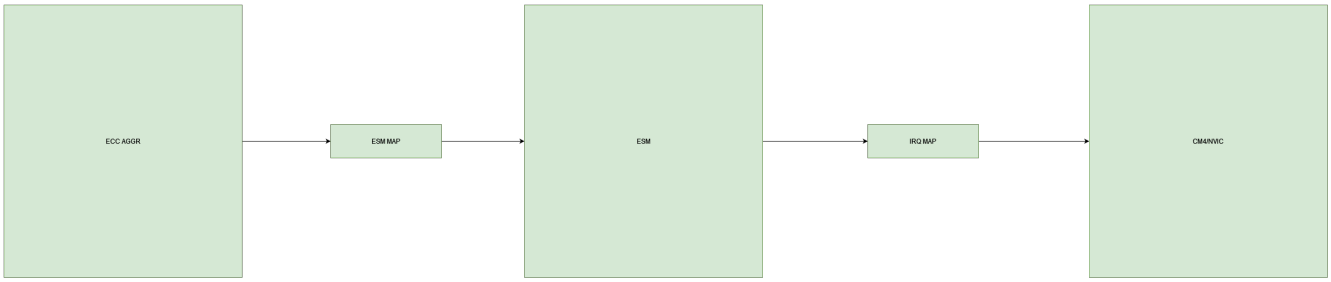
Table 22-59. DCCERRCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	NU15	R	0h	Reserved
9-0	ERRCNT	R/W	0h	Counts the number of errors after the last write to this register or reset. If reached terminal count the count freezes. User needs to clear it.

22.3 ECC Aggregator

22.3.1 Overview

To increase functional safety and system reliability the memories (for example, FIFOs, queues, SRAMs and others) in many device modules and subsystems are protected by error correcting code (ECC). This is accomplished through an ECC aggregator and ECC wrapper. The ECC aggregator is connected to these memories (hereinafter ECC RAMs) and involved in the ECC process. Each memory is surrounded by an ECC wrapper which performs the ECC detection and correction. The wrapper communicates via serial interface with the aggregator which has memory mapped configuration interface.



22.3.2 IP Design Info

22.3.2.1 ECC Aggregator Features

The ECC aggregator has the following features:

- Reduces memory software errors via single error correction (SEC) and double error detection (DED)
- Provides a mechanism to control and monitor the ECC protected memories in a module or subsystem
- SEC and DED over the system interconnect data bus and parity and redundancy for the system interconnect command and address buses
- Generates an interrupt for correctable error
- Generates an interrupt for non-correctable error
- Supports inject only mode for diagnostic purposes
- Supports software readable status for single and double-bit ECC errors and associated information such as row address where error has occurred and data bits that have been flipped
- An ECC endpoint can be ECC RAM component.
- Detects single bit error via parity checking on:
  1. Memory mapped configuration interface FIFO
  2. Serial interface FIFO
- Single bit error detection via parity checking results in a non-correctable error interrupt
- Supports timeout mechanism on transactions over the ECC serial interface. Timeout occurrence results in a non-correctable error interrupt.
- Certain control bits have redundancy and if a bit flips an interrupt is generated

22.3.2.2 ECC Aggregator Integration

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

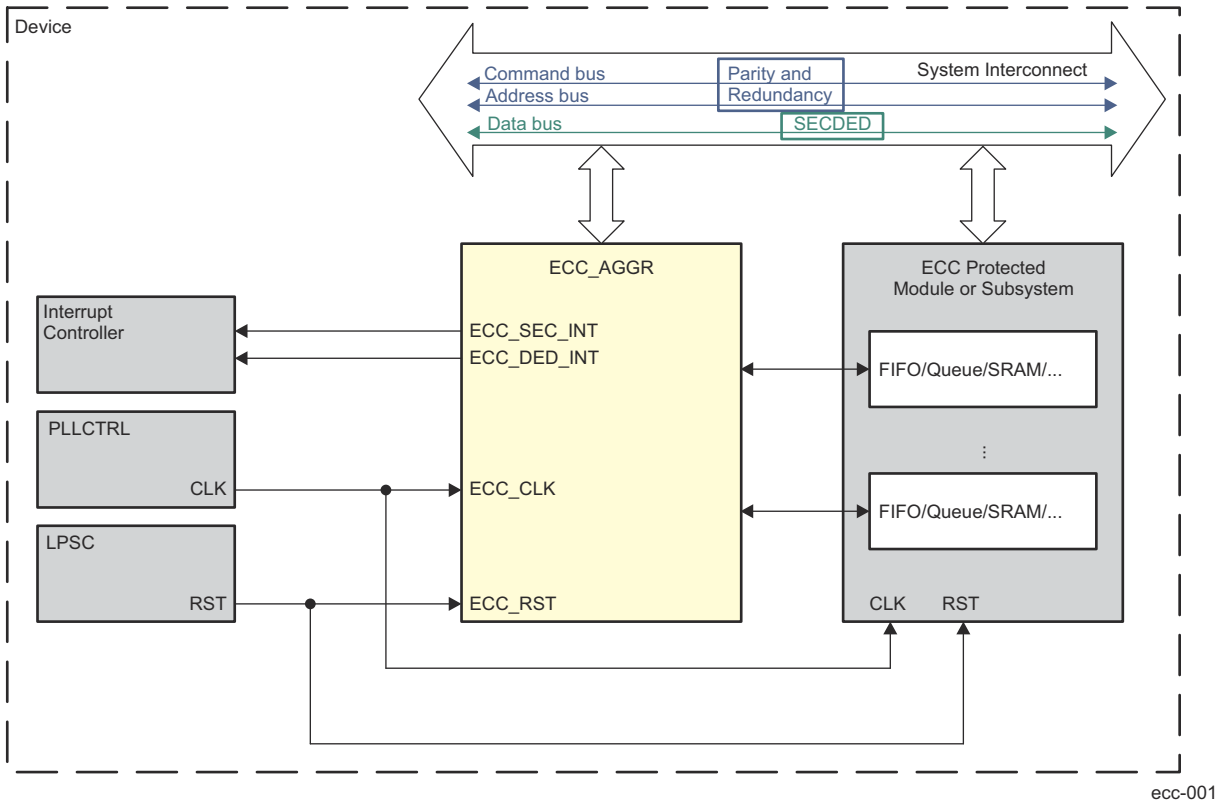


Figure 22-14. ECC Aggregator Integration

Table 22-60. ECC Aggregator Clocks and Resets

Clock				
Module Instance	Module Clock Input	Source Clock Signal	Source	Description
ECC_AGGR	ECC_CLK	Same as corresponding module or subsystem	Same as corresponding module or subsystem	ECC aggregator clock
Resets				
Module Instance	Module Reset Input	Source Reset Signal	Source	Description
ECC_AGGR	ECC_RST	Same as corresponding module or subsystem	Same as corresponding module or subsystem	ECC aggregator reset

Table 22-61. ECC Aggregator Hardware Requests

Interrupt Requests					
Module Instance	Module Interrupt Signal	Destination Interrupt Input	Description	Description	Type
ECC_AGGR	ECC_SEC_INT	See	See	Interrupt for correctable error(SEC)	Leve
	ECC_DED_INT	See	See	Interrupt for non-correctable error (DED, parity, redundancy, timeout)	LEVEL
DMA Events					
Module Instance	Module DMA Input	Destination DMA Event Input	Destination	Description	Type
ECC_AGGR	-	-	-	-	-

ADVANCE INFORMATION

Note

For more information on the interrupts, see Section 22.3.2.7.

For more information on the interconnects, see Section 3.

For more information on the power, reset and clock management, see the corresponding sections in Section 5.

For more information on the device interrupt controllers, see *Interrupt Controllers*.

22.3.2.3 ECC Aggregator Function Description

This section describes the architecture and functional details of the ECC aggregator.

22.3.2.3.1 ECC Aggregator Block Diagram

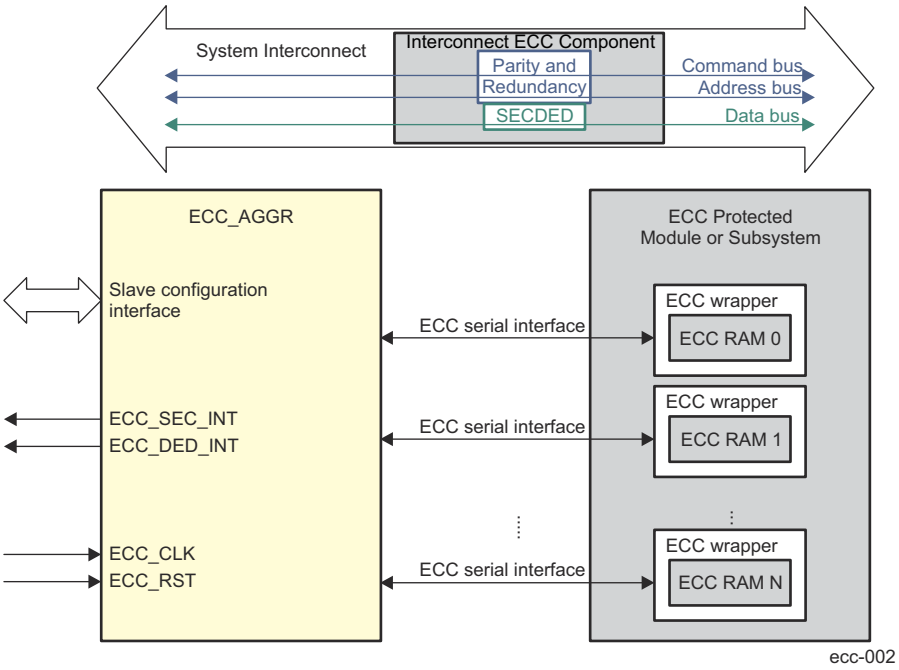


Figure 22-15. ECC Aggregator Block Diagram

The ECC aggregator is connected to one or more ECC endpoints each of which has assigned a unique ID used when the endpoint is accessed for status information or configuration. The ECC aggregator provides software access to all ECC related registers through its memory mapped slave configuration interface while the serial interface is used to communicate with the ECC endpoints. Upon detection of single or double-bit error the corresponding interrupt line is asserted.

22.3.2.4 ECC Aggregator Register Groups

The ECC aggregator has ECC control, status and interrupt registers for each ECC endpoint in a module or subsystem. These registers are memory mapped and occupy 1 KB address space although part of it may contain reserved locations. The registers are split in the following types:

- **Global registers**. They are common to all ECC endpoints associated with the ECC aggregator and include the ECC_VECTOR and ECC_REV registers. Each ECC endpoint has assigned a unique ID.

  When this ID is written to the ECC_VECTOR[10-0] ECC_VECTOR field the corresponding endpoint is selected either for control or for status reading.
- **ECC control and status registers**. These registers are specific to each ECC endpoint and reside in the range from address offset 0x10 to 0x28, if the endpoint is ECC RAM or from 0x10 to 0x24, if the endpoint is interconnect ECC component. They are memory mapped but are accessed through the ECC serial interface. They are also selected by the ECC endpoint ID written to the ECC_VECTOR[10-0] ECC_VECTOR field.

Because of latency on the serial interface the ECC control and status registers are read by performing special sequence as described in Section 12.9.4.3.3. These registers have also different functionality for both types of endpoints - ECC RAM and interconnect ECC component.

- **Interrupt registers**. They include interrupt status, interrupt enable, interrupt disable, and EOI registers.

22.3.2.5 Read Access to the ECC Control and Status Registers

Read accesses to the ECC control and status registers for each ECC endpoint represent read operations over the ECC serial interface and are triggered by performing the following sequence:

1. Software writes the following in the ECC_VECTOR register:
   - The ECC endpoint ID in the ECC_VECTOR[10-0] ECC_VECTOR field to select particular ECC endpoint.
   - The register read address in the ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field to select which register has to be read through the ECC serial interface.
   - A value of 0x1 in the ECC_VECTOR[15] RD_SVBUS bit to trigger read operation through the ECC serial interface.
2. Software polls the ECC_VECTOR[24] RD_SVBUS_DONE bit to check if it is 0x1. This indicates that the read operation on the ECC serial interface has completed.
3. Software reads the data from the register previously selected by the ECC_VECTOR[23-16]RD_SVBUS_ADDRESS field.

22.3.2.6 Serial Write Operation

Write operations over the ECC serial interface are performed as follows:

1. Software specifies the ECC endpoint ID in the ECC_VECTOR[10-0] ECC_VECTOR field. The ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field is a don't care but the ECC_VECTOR[15] RD_SVBUS bit must be set to 0x0.
2. Software performs regular write operation to the desired address. If the ECC endpoint ID has already been specified, step 1 can be skipped. Unlike serial read operations it is not necessary to always specify the endpoint ID before performing serial write operation.

The following is an example for serial write operation:

1. Write 0x0000 0008 to the ECC_VECTOR register.
2. Write 0x0000 000F to the ECC_CTRL register. This sends write request with data 0x0000 000F to the ECC_CTRL register associated with ECC RAM with ID = 8.

22.3.2.7 Interrupts

The ECC aggregator generates the following interrupts:

- Correctable interrupt (ECC_SEC_INT) where hardware can correct the error but notifies the system in case of SEC.
- Non-correctable interrupt (ECC_DED_INT) where hardware cannot correct the error in cases of DED, parity check, redundancy check or timeout occurrence.

The following is the sequence for servicing interrupts:

- Software enables the interrupts for an ECC endpoint by writing 0x1 to the corresponding bit of the following interrupt enable registers:
  - ECC_SEC_ENABLE_SET_REG0 for the correctable interrupt
  - ECC_DED_ENABLE_SET_REG0 for the noncorrectable interrupt
- On receiving an interrupt, software checks which ECC endpoint has caused the error by reading the following interrupt status registers:
  - ECC_SEC_STATUS_REG0 for the correctable interrupt ECC_DED_STATUS_REG0 for the non-correctable interrupt
- Software performs serial read operations as described in Section 12.9.4.3.3 to read the following status registers that contain details about the error:
  - If the endpoint is ECC RAM:
    - ECC_ERR_STAT1
    - ECC_ERR_STAT2

    - ECC_ERR_STAT3
- If the endpoint is interconnect ECC component:
  - ECC_CBASS_ERR_STAT1
  - ECC_CBASS_ERR_STAT2
- After the interrupt has been serviced, depending on the error type, software should clear the corresponding status bits in the ECC_ERR_STAT1 and ECC_ERR_STAT3 registers or in the ECC_CBASS_ERR_STAT1 register. Software has to poll these registers to guarantee that status bits are cleared as there is no other indication for write completion over the ECC serial interface.

The value of the *_PEND_CLR fields in the ECC_CBASS_ERR_STAT1 register must be read and then written back to decrement the count of each field back to 0x0. A further error capture into the ECC_CBASS_ERR_STAT1 register does not occur unless all its fields are 0x0. The decrement value should not be larger than the read value. If a field in the ECC_CBASS_ERR_STAT1 register should not be modified, write a value of 0x0 to that field.

- Software writes 0x1 to the corresponding end of interrupt register to clear the interrupt:
  - ECC_SEC_EOI_REG for the correctable interrupt
  - ECC_DED_EOI_REG for the non-correctable interrupt

22.3.2.8 Inject Only Mode

There are modules that already perform the ECC generation and checking as part of their data path. In this case, the ECC wrapper may be configured in inject only mode, if needed. In this mode the ECC wrapper does not perform ECC detection and correction. The inject only mode allows users to inject single or double-bit errors so that the module logic can be tested for diagnostic purposes.

Note

There is no software control to enable inject only mode. It is configured via tie-off value. Inject only and ECC modes are mutually exclusive.

The interconnect ECC component also supports error injection mode. There is error injection logic for testing of the error checking logic (checkers). The injection logic can be configured to inject either single or double bit error and what data pattern to be used for injection (ECC_CBASS_CTRL[11-8] ECC_PATTERN). The ECC_CBASS_ERR_CTRL1 and ECC_CBASS_ERR_CTRL2 registers should be written first to setup the injection. Then, either the ECC_CBASS_CTRL[3] FORCE_SE or the ECC_CBASS_CTRL[4] FORCE_DE bit must be set to 0x1 to start the injection. Both bits must not be set at the same time. If the injection should continue in incrementing mode, then the ECC_CBASS_CTRL[5] FORCE_N_BIT bit should be set to 0x1. Once the FORCE_N_BIT is set, then each successive injection can simply write the ECC_CBASS_CTRL register to set the FORCE_SE or FORCE_DE again. Reading 0x0 from either the FORCE_SE or the FORCE_DE bit indicates that the injection has completed, as these bits automatically clear when the checker indicates that it has performed the injection. The time for an injection to complete is not guaranteed, so some delay is needed between successive injections.

22.3.2.9 Errors

Each aggregator generates two errors which drive the ESM.

- <modulename>_SERR module names are mentioned in the below section
- <modulename>_SERR module names are mentioned in the below section

Group1 and Group2 mappings are found the ESM interrupt sections.

22.3.3 Details

To increase functional safety and system reliability the memories (for example, FIFOs, queues, SRAMs and others) in many device modules and subsystems are protected by error correcting code (ECC). There are two ecc aggregators in xWRLx432:

- APP_SS_ECC_AGG in APPSS
- FECSS_CM3_ECC_AGGR in FECSS

This Aggregator is used to fault inject all memory ecc_controllers and also aggregate the errors to generate a single error to ESM.

22.3.4 Interupts and Errors

Each Aggregator generates total of two Errors which would be driving the ESM.

This device has memories that has ECC and some memories have Parity and some memories are not protected. Refer to "MEMORY INITIALIZATION" regarding the initialization of memories protected with ECC or Parity.

Instance	Error Intrerrupts
APP_SS_ECC_AGG	appss_cm4f_ram_ecc_agg_serr appss_cm4f_ram_ecc_agg_uerr

22.3.5 Aggregator Mapping to Memory Instances

Table 22-62. RAMID Information for Each ECC Aggregator:

Memory Bank	RAMID	ECC Aggregator
APP_SS_ROM	0	APP_ECC_AGG Registers
APP_SS_RAM1	1	
APP_SS_RAM2	2	
APP_SS_RAM3	3	
APP_SS_TPTC1	4	
APP_SS_TPTC2	5	
HWA_TPTC1	6	
HWA_TPTC2	7	
SHARED_MEM0	8	
SHARED_MEM1	9	
HWA_PARAM_MEM	10	
SHARED_MEM2	11	
ADCPING_MEM	12	
ADCPONG_MEM	13	

22.3.6 APP_ECC_AGG Registers

Table 22-63 lists the memory-mapped registers for the APP_ECC_AGG registers. All register offset addresses not listed in Table 22-63 should be considered as reserved locations and the register contents should not be modified.

Table 22-63. APP_ECC_AGG Registers

Offset	Acronym	Register Name	Section
0h	AGGR_REVISION	AGGR_REVISION	Go
8h	ECC_VECTOR	ECC_VECTOR	Go
Ch	MISC_STATUS	MISC_STATUS	Go
10h	ECC_WRAP_REVISION	ECC_WRAP_REVISION	Go
14h	CONTROL	CONTROL	Go
18h	ERROR_CTRL1	ERROR_CTRL1	Go
1Ch	ERROR_CTRL2	ERROR_CTRL2	Go
20h	ERROR_STATUS1	ERROR_STATUS1	Go
24h	ERROR_STATUS2	ERROR_STATUS2	Go
28h	ERROR_STATUS3	ERROR_STATUS3	Go
3Ch	SEC_EOI_REG	SEC_EOI_REG	Go
40h	SEC_STATUS_REG0	SEC_STATUS_REG0	Go
80h	SEC_ENABLE_SET_REG0	SEC_ENABLE_SET_REG0	Go
C0h	SEC_ENABLE_CLR_REG0	SEC_ENABLE_CLR_REG0	Go
13Ch	DED_EOI_REG	DED_EOI_REG	Go
140h	DED_STATUS_REG0	DED_STATUS_REG0	Go
180h	DED_ENABLE_SET_REG0	DED_ENABLE_SET_REG0	Go
1C0h	DED_ENABLE_CLR_REG0	DED_ENABLE_CLR_REG0	Go
200h	AGGR_ENABLE_SET	AGGR_ENABLE_SET	Go
204h	AGGR_ENABLE_CLR	AGGR_ENABLE_CLR	Go
208h	AGGR_STATUS_SET	AGGR_STATUS_SET	Go
20Ch	AGGR_STATUS_CLR	AGGR_STATUS_CLR	Go

Complex bit access types are encoded to fit into small table cells. Table 22-64 shows the codes that are used for access types in this section.

Table 22-64. APP_ECC_AGG 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

22.3.6.1 AGGR_REVISION Register (Offset = 0h) [Reset = 66A0EA00h]

AGGR_REVISION is shown in Table 22-65.

Return to the Summary Table.

Revision parameters

Table 22-65. AGGR_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Scheme - (RO)
29-28	BU	R	2h	bu - (RO)
27-16	MODULE_ID	R	6A0h	Module ID - (RO)
15-11	REVRTL	R	1Dh	RTL version - (RO)
10-8	REVMAJ	R	2h	Major version - (RO)
7-6	CUSTOM	R	0h	Custom version - (RO)
5-0	REVMIN	R	0h	Minor version - (RO)

22.3.6.2 ECC_VECTOR Register (Offset = 8h) [Reset = 00000000h]

ECC_VECTOR is shown in Table 22-66.

Return to the Summary Table.

ECC Vector Register

Table 22-66. ECC_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RES1	R	0h	RESERVE FIELD
24	RD_SVBUS_DONE	R	0h	Status to indicate if read on serial VBUS is complete - (RO)
23-16	RD_SVBUS_ADDRESS	R/W	0h	Read address - (RW)
15	RD_SVBUS	R/W	0h	Write 1 to trigger a read on the serial VBUS - (RW)
14-11	RES2	R	0h	RESERVE FIELD
10-0	ECC_VECTOR	R/W	0h	Value written to select the corresponding ECC RAM for control or status - (RW)

22.3.6.3 MISC_STATUS Register (Offset = Ch) [Reset = 0000000Eh]

MISC_STATUS is shown in Table 22-67.

Return to the Summary Table.

Misc Status

Table 22-67. MISC_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RES3	R	0h	RESERVE FIELD
10-0	NUM_RAMS	R	Eh	Indicates the number of RAMS serviced by the ECC aggregator - (RO)

22.3.6.4 ECC_WRAP_REVISION Register (Offset = 10h) [Reset = 66A40202h]

ECC_WRAP_REVISION is shown in Table 22-68.

Return to the Summary Table.

Revision parameters

Table 22-68. ECC_WRAP_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	Scheme - (RO)
29-28	BU	R	2h	bu - (RO)
27-16	MODULE_ID	R	6A4h	Module ID - (RO)
15-11	REVRTL	R	0h	RTL version - (RO)

Table 22-68. ECC_WRAP_REVISION Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	REVMAJ	R	2h	Major version - (RO)
7-6	CUSTOM	R	0h	Custom version - (RO)
5-0	REVMIN	R	2h	Minor version - (RO)

22.3.6.5 CONTROL Register (Offset = 14h) [Reset = 00000187h]

CONTROL is shown in Table 22-69.

Return to the Summary Table.

ECC Control Register

Table 22-69. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RES4	R	0h	RESERVE FIELD
8	CHECK_SVBUS_TIMEOUT	R/W	1h	check for svbus timeout errors - (RW)
7	CHECK_PARITY	R/W	1h	check for parity errors - (RW)
6	ERROR_ONCE	R/W	0h	Force Error only once - (RW)
5	FORCE_N_ROW	R/W	0h	Force Error on any RAM read - (RW)
4	FORCE_DED	R/W	0h	Force Double Bit Error - (RW)
3	FORCE_SEC	R/W	0h	Force Single Bit Error - (RW)
2	ENABLE_RMW	R/W	1h	Enable rmw - (RW)
1	ECC_CHECK	R/W	1h	Enable ECC check - (RW)
0	ECC_ENABLE	R/W	1h	Enable ECC - (RW)

22.3.6.6 ERROR_CTRL1 Register (Offset = 18h) [Reset = 00000000h]

ERROR_CTRL1 is shown in Table 22-70.

Return to the Summary Table.

ECC Error Control1 Register

Table 22-70. ERROR_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	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 - (RW)

22.3.6.7 ERROR_CTRL2 Register (Offset = 1Ch) [Reset = 00000000h]

ERROR_CTRL2 is shown in Table 22-71.

Return to the Summary Table.

ECC Error Control2 Register

Table 22-71. ERROR_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT2	R/W	0h	Data bit that needs to be flipped if double bit error needs to be forced - (RW)
15-0	ECC_BIT1	R/W	0h	Data bit that needs to be flipped when force_sec is set - (RW)

22.3.6.8 ERROR_STATUS1 Register (Offset = 20h) [Reset = 00000000h]

ERROR_STATUS1 is shown in Table 22-72.

Return to the Summary Table.

ECC Error Status1 Register

Table 22-72. ERROR_STATUS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT1	R	0h	Data bit that corresponds to the single-bit error - (RO)
15	CLR_CTRL_REG_ERR	R/W	0h	Clear control reg error Error Status you must also re write the control ergister itself to clear this - (RW)
14-13	CLR_PARITY_ERR	R/W	0h	Clear parity Error Status - (RW decr)
12	CLR_ECC_OTHER	R/W	0h	Clear other Error Status - (RW)
11-10	CLR_ECC_DED	R/W	0h	Clear Double Bit Error Status - (RW decr)
9-8	CLR_ECC_SEC	R/W	0h	Clear Single Bit Error Status - (RW decr)
7	CTR_REG_ERR	R/W	0h	control register error pending Level interrupt - (RW
6-5	PARITY_ERR	R/W	0h	Level parity error Error Status - (RW)
4	ECC_OTHER	R/W	0h	successive single-bit errors have occurred while a writeback is still pending Level interrupt - (RW
3-2	ECC_DED	R/W	0h	Level Double Bit Error Status - (RW incr)
1-0	ECC_SEC	R/W	0h	Level Single Bit Error Status - (RW incr)

22.3.6.9 ERROR_STATUS2 Register (Offset = 24h) [Reset = 00000000h]

ERROR_STATUS2 is shown in Table 22-73.

Return to the Summary Table.

ECC Error Status2 Register

Table 22-73. ERROR_STATUS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECC_ROW	R	0h	Row address where the single or double-bit error has occurred - (RO)

22.3.6.10 ERROR_STATUS3 Register (Offset = 28h) [Reset = 00000000h]

ERROR_STATUS3 is shown in Table 22-74.

Return to the Summary Table.

ECC Error Status3 Register

Table 22-74. ERROR_STATUS3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RES5	R	0h	RESERVE FIELD
9	CLR_SVBUS_TIMEOUT_ERR	R/W	0h	Clear svbus timeout Error Status - (RW)
8-2	RES6	R	0h	RESERVE FIELD
1	SVBUS_TIMEOUT_ERR	R/W	0h	Level svbus timeout error Error Status - (RW)
0	WB_PEND	R	0h	delayed write back pending Status - (RO)

22.3.6.11 SEC_EOI_REG Register (Offset = 3Ch) [Reset = 00000000h]

SEC_EOI_REG is shown in Table 22-75.

Return to the Summary Table.

EOI Register

Table 22-75. SEC_EOI_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RES7	R	0h	RESERVE FIELD
0	EOI_WR	R/W	0h	EOI Register - (RW)

22.3.6.12 SEC_STATUS_REG0 Register (Offset = 40h) [Reset = 00000000h]

SEC_STATUS_REG0 is shown in Table 22-76.

Return to the Summary Table.

Interrupt Status Register 0

Table 22-76. SEC_STATUS_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES8	R	0h	RESERVE FIELD
13	ADCPONG_RAMECC_PEND	R/W	0h	Interrupt Pending Status for adcpong_ramecc_pend - (RW)
12	ADCPING_RAMECC_PEND	R/W	0h	Interrupt Pending Status for adcping_ramecc_pend - (RW)
11	SHARED_MEM2_RAMECC_PEND	R/W	0h	Interrupt Pending Status for shared_mem2_ramecc_pend - (RW)
10	HWA_PARAM_MEM_PEND	R/W	0h	Interrupt Pending Status for hwa_param_mem_pend - (RW)
9	SHARED_MEM1_RAMECC_PEND	R/W	0h	Interrupt Pending Status for shared_mem1_ramecc_pend - (RW)
8	SHARED_MEM0_RAMECC_PEND	R/W	0h	Interrupt Pending Status for shared_mem0_ramecc_pend - (RW)
7	HWA_TPTC2_PEND	R/W	0h	Interrupt Pending Status for hwa_tptc2_pend - (RW)
6	HWA_TPTC1_PEND	R/W	0h	Interrupt Pending Status for hwa_tptc1_pend - (RW)
5	APP_SS_TPTC2_PEND	R/W	0h	Interrupt Pending Status for app_ss_tptc2_pend - (RW)
4	APP_SS_TPTC1_PEND	R/W	0h	Interrupt Pending Status for app_ss_tptc1_pend - (RW)
3	APP_SS_RAM3_PEND	R/W	0h	Interrupt Pending Status for app_ss_ram3_pend - (RW)
2	APP_SS_RAM2_PEND	R/W	0h	Interrupt Pending Status for app_ss_ram2_pend - (RW)
1	APP_SS_RAM1_PEND	R/W	0h	Interrupt Pending Status for app_ss_ram1_pend - (RW)
0	APP_SS_ROM_PEND	R/W	0h	Interrupt Pending Status for app_ss_rom_pend - (RW)

22.3.6.13 SEC_ENABLE_SET_REG0 Register (Offset = 80h) [Reset = 00000000h]

SEC_ENABLE_SET_REG0 is shown in Table 22-77.

Return to the Summary Table.

Interrupt Enable Set Register 0

Table 22-77. SEC_ENABLE_SET_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES9	R	0h	RESERVE FIELD
13	ADCPONG_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for adcpong_ramecc_pend - (RW)
12	ADCPING_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for adcping_ramecc_pend - (RW)

Table 22-77. SEC_ENABLE_SET_REG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SHARED_MEM2_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for shared_mem2_ramecc_pend - (RW)
10	HWA_PARAM_MEM_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for hwa_param_mem_pend - (RW)
9	SHARED_MEM1_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for shared_mem1_ramecc_pend - (RW)
8	SHARED_MEM0_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for shared_mem0_ramecc_pend - (RW)
7	HWA_TPTC2_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for hwa_tptc2_pend - (RW)
6	HWA_TPTC1_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for hwa_tptc1_pend - (RW)
5	APP_SS_TPTC2_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_tptc2_pend - (RW)
4	APP_SS_TPTC1_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_tptc1_pend - (RW)
3	APP_SS_RAM3_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_ram3_pend - (RW)
2	APP_SS_RAM2_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_ram2_pend - (RW)
1	APP_SS_RAM1_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_ram1_pend - (RW)
0	APP_SS_ROM_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_rom_pend - (RW)

22.3.6.14 SEC_ENABLE_CLR_REG0 Register (Offset = C0h) [Reset = 00000000h]

SEC_ENABLE_CLR_REG0 is shown in Table 22-78.

Return to the Summary Table.

Interrupt Enable Clear Register 0

Table 22-78. SEC_ENABLE_CLR_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES10	R	0h	RESERVE FIELD
13	ADCPONG_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for adcpong_ramecc_pend - (RW)
12	ADCPING_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for adcping_ramecc_pend - (RW)
11	SHARED_MEM2_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for shared_mem2_ramecc_pend - (RW)
10	HWA_PARAM_MEM_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for hwa_param_mem_pend - (RW)
9	SHARED_MEM1_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for shared_mem1_ramecc_pend - (RW)
8	SHARED_MEM0_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for shared_mem0_ramecc_pend - (RW)
7	HWA_TPTC2_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for hwa_tptc2_pend - (RW)
6	HWA_TPTC1_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for hwa_tptc1_pend - (RW)
5	APP_SS_TPTC2_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_tptc2_pend - (RW)
4	APP_SS_TPTC1_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_tptc1_pend - (RW)

Table 22-78. SEC_ENABLE_CLR_REG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	APP_SS_RAM3_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_ram3_pend - (RW)
2	APP_SS_RAM2_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_ram2_pend - (RW)
1	APP_SS_RAM1_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_ram1_pend - (RW)
0	APP_SS_ROM_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_rom_pend - (RW)

22.3.6.15 DED_EOI_REG Register (Offset = 13Ch) [Reset = 00000000h]

DED_EOI_REG is shown in Table 22-79.

Return to the Summary Table.

EOI Register

Table 22-79. DED_EOI_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RES11	R	0h	RESERVE FIELD
0	EOI_WR	R/W	0h	EOI Register - (RW)

22.3.6.16 DED_STATUS_REG0 Register (Offset = 140h) [Reset = 00000000h]

DED_STATUS_REG0 is shown in Table 22-80.

Return to the Summary Table.

Interrupt Status Register 0

Table 22-80. DED_STATUS_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES12	R	0h	RESERVE FIELD
13	ADCPONG_RAMECC_PEND	R/W	0h	Interrupt Pending Status for adcpong_ramecc_pend - (RW)
12	ADCPING_RAMECC_PEND	R/W	0h	Interrupt Pending Status for adcping_ramecc_pend - (RW)
11	SHARED_MEM2_RAMECC_PEND	R/W	0h	Interrupt Pending Status for shared_mem2_ramecc_pend - (RW)
10	HWA_PARAM_MEM_PEND	R/W	0h	Interrupt Pending Status for hwa_param_mem_pend - (RW)
9	SHARED_MEM1_RAMECC_PEND	R/W	0h	Interrupt Pending Status for shared_mem1_ramecc_pend - (RW)
8	SHARED_MEM0_RAMECC_PEND	R/W	0h	Interrupt Pending Status for shared_mem0_ramecc_pend - (RW)
7	HWA_TPTC2_PEND	R/W	0h	Interrupt Pending Status for hwa_tptc2_pend - (RW)
6	HWA_TPTC1_PEND	R/W	0h	Interrupt Pending Status for hwa_tptc1_pend - (RW)
5	APP_SS_TPTC2_PEND	R/W	0h	Interrupt Pending Status for app_ss_tptc2_pend - (RW)
4	APP_SS_TPTC1_PEND	R/W	0h	Interrupt Pending Status for app_ss_tptc1_pend - (RW)
3	APP_SS_RAM3_PEND	R/W	0h	Interrupt Pending Status for app_ss_ram3_pend - (RW)
2	APP_SS_RAM2_PEND	R/W	0h	Interrupt Pending Status for app_ss_ram2_pend - (RW)
1	APP_SS_RAM1_PEND	R/W	0h	Interrupt Pending Status for app_ss_ram1_pend - (RW)
0	APP_SS_ROM_PEND	R/W	0h	Interrupt Pending Status for app_ss_rom_pend - (RW)

22.3.6.17 DED_ENABLE_SET_REG0 Register (Offset = 180h) [Reset = 00000000h]

DED_ENABLE_SET_REG0 is shown in Table 22-81.

Return to the Summary Table.

Interrupt Enable Set Register 0

Table 22-81. DED_ENABLE_SET_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES13	R	0h	RESERVE FIELD
13	ADCPONG_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for adcpong_ramecc_pend - (RW)
12	ADCPING_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for adcping_ramecc_pend - (RW)
11	SHARED_MEM2_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for shared_mem2_ramecc_pend - (RW)
10	HWA_PARAM_MEM_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for hwa_param_mem_pend - (RW)
9	SHARED_MEM1_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for shared_mem1_ramecc_pend - (RW)
8	SHARED_MEM0_RAMECC_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for shared_mem0_ramecc_pend - (RW)
7	HWA_TPTC2_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for hwa_tptc2_pend - (RW)
6	HWA_TPTC1_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for hwa_tptc1_pend - (RW)
5	APP_SS_TPTC2_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_tptc2_pend - (RW)
4	APP_SS_TPTC1_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_tptc1_pend - (RW)
3	APP_SS_RAM3_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_ram3_pend - (RW)
2	APP_SS_RAM2_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_ram2_pend - (RW)
1	APP_SS_RAM1_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_ram1_pend - (RW)
0	APP_SS_ROM_ENABLE_SET	R/W	0h	Interrupt Enable Set Register for app_ss_rom_pend - (RW)

22.3.6.18 DED_ENABLE_CLR_REG0 Register (Offset = 1C0h) [Reset = 00000000h]

DED_ENABLE_CLR_REG0 is shown in Table 22-82.

Return to the Summary Table.

Interrupt Enable Clear Register 0

Table 22-82. DED_ENABLE_CLR_REG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RES14	R	0h	RESERVE FIELD
13	ADCPONG_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for adcpong_ramecc_pend - (RW)
12	ADCPING_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for adcping_ramecc_pend - (RW)
11	SHARED_MEM2_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for shared_mem2_ramecc_pend - (RW)
10	HWA_PARAM_MEM_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for hwa_param_mem_pend - (RW)

Table 22-82. DED_ENABLE_CLR_REG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SHARED_MEM1_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for shared_mem1_ramecc_pend - (RW)
8	SHARED_MEM0_RAMECC_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for shared_mem0_ramecc_pend - (RW)
7	HWA_TPTC2_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for hwa_tptc2_pend - (RW)
6	HWA_TPTC1_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for hwa_tptc1_pend - (RW)
5	APP_SS_TPTC2_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_tptc2_pend - (RW)
4	APP_SS_TPTC1_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_tptc1_pend - (RW)
3	APP_SS_RAM3_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_ram3_pend - (RW)
2	APP_SS_RAM2_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_ram2_pend - (RW)
1	APP_SS_RAM1_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_ram1_pend - (RW)
0	APP_SS_ROM_ENABLE_CLR	R/W	0h	Interrupt Enable Clear Register for app_ss_rom_pend - (RW)

22.3.6.19 AGGR_ENABLE_SET Register (Offset = 200h) [Reset = 00000000h]

AGGR_ENABLE_SET is shown in Table 22-83.

Return to the Summary Table.

AGGR interrupt enable set Register

Table 22-83. AGGR_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES15	R	0h	RESERVE FIELD
1	TIMEOUT	R/W	0h	interrupt enable set for svbus timeout errors - (RW)
0	PARITY	R/W	0h	interrupt enable set for parity errors - (RW)

22.3.6.20 AGGR_ENABLE_CLR Register (Offset = 204h) [Reset = 00000000h]

AGGR_ENABLE_CLR is shown in Table 22-84.

Return to the Summary Table.

AGGR interrupt enable clear Register

Table 22-84. AGGR_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RES16	R	0h	RESERVE FIELD
1	TIMEOUT	R/W	0h	interrupt enable clear for svbus timeout errors - (RW)
0	PARITY	R/W	0h	interrupt enable clear for parity errors - (RW)

22.3.6.21 AGGR_STATUS_SET Register (Offset = 208h) [Reset = 00000000h]

AGGR_STATUS_SET is shown in Table 22-85.

Return to the Summary Table.

AGGR interrupt status set Register

Table 22-85. AGGR_STATUS_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RES17	R	0h	RESERVE FIELD
3-2	TIMEOUT	R/W	0h	interrupt status set for svbus timeout errors - (RW incr)
1-0	PARITY	R/W	0h	interrupt status set for parity errors - (RW incr)

22.3.6.22 AGGR_STATUS_CLR Register (Offset = 20Ch) [Reset = 00000000h]

AGGR_STATUS_CLR is shown in Table 22-86.

Return to the Summary Table.

AGGR interrupt status clear Register

Table 22-86. AGGR_STATUS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RES18	R	0h	RESERVE FIELD
3-2	TIMEOUT	R/W	0h	interrupt status clear for svbus timeout errors - (RW decr)
1-0	PARITY	R/W	0h	interrupt status clear for parity errors - (RW decr)

22.4 ESM

22.4.1 Overview

The Error Signaling Module (ESM) aggregates safety-related events and/or errors from throughout the device into one location. It can signal both low and high priority interrupts to a processor to deal with a safety event and/or manipulate an I/O error pin to signal an external hardware that an error has occurred. Therefore an external controller is able to reset the device or keep the system in a safe, known state.

22.4.2 Feature List

- Up to 40 error channels (APPSS) are supported, divided into 3 different groups:
  - 32 Group1 (low severity) channels with configurable interrupt generation and configurable $\overline{\text{ERROR}}$ pin behavior
  - 8 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

22.4.3 ESM Block Diagram

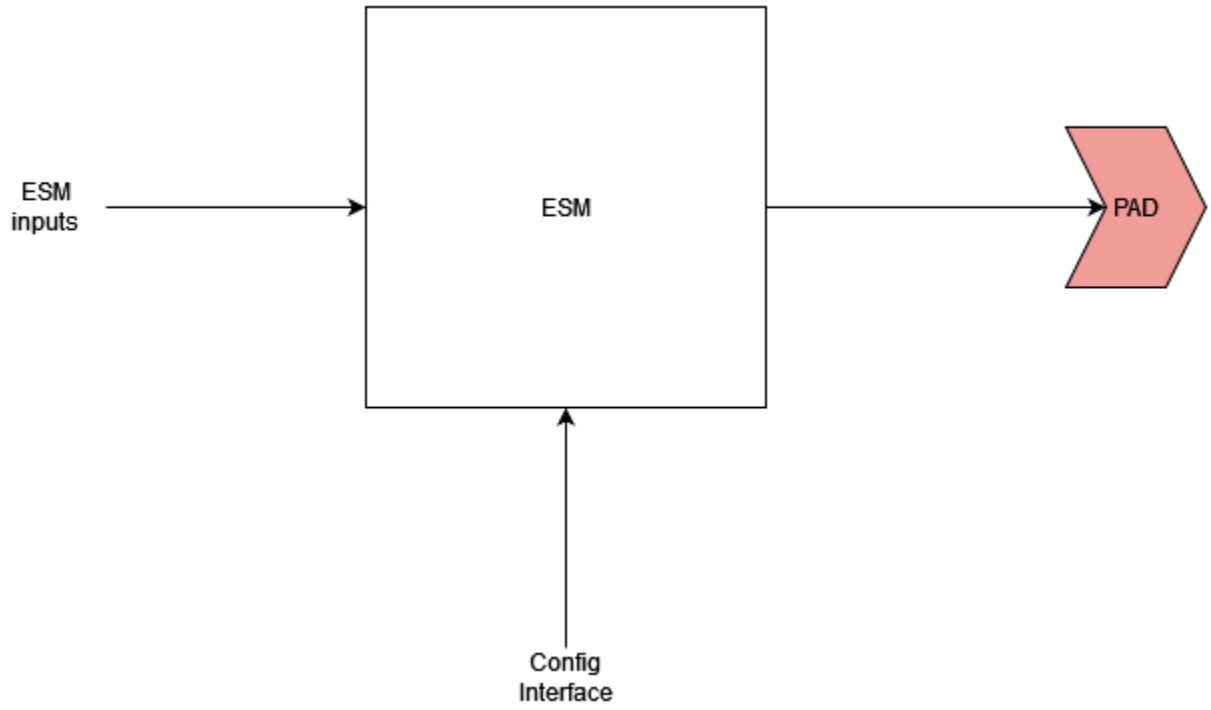


Figure 22-16. ESM Block Diagram

22.4.4 Module Operation

This device has 40 error channels (APPSS), divided into 2 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 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. 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 $\overline{\text{RST}}$.

The functionality of the $\overline{\text{ERROR}}$ pin can be tested by forcing an error.

22.4.4.1 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.

22.4.4.2 $\overline{\text{ERROR}}$ 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:

$$t_{\overline{ERROR}\_low} = t_{VCLK} \times (LTCP + 1) \tag{9}$$

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.

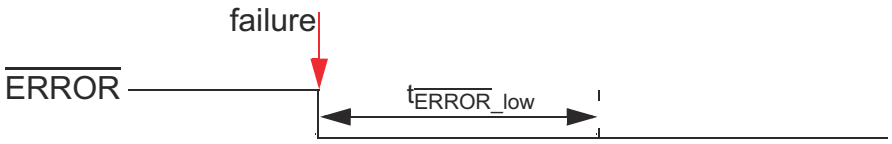


Figure 22-17. $\overline{\text{ERROR}}$ Pin Timing - Example 1

Example 2: ESM detects a failure and drives the $\overline{\text{ERROR}}$ pin low. An $\overline{\text{ERROR}}$ pin reset request is received before $t_{\overline{\text{ERROR}}\_low}$ expires. In this case, the $\overline{\text{ERROR}}$ pin is set to high immediately after $t_{\overline{\text{ERROR}}\_low}$ expires.

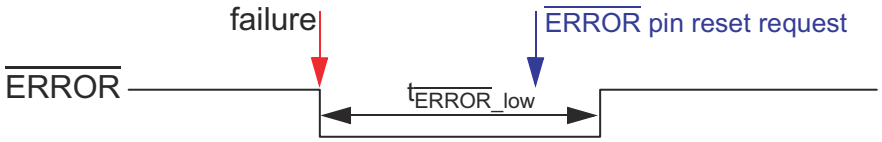


Figure 22-18. $\overline{\text{ERROR}}$ Pin Timing - Example 2

Example 3: ESM detects a failure and drives the $\overline{\text{ERROR}}$ pin low. An $\overline{\text{ERROR}}$ pin reset request is received after $t_{\overline{\text{ERROR}}\_low}$ expires. In this case, the $\overline{\text{ERROR}}$ pin is set to high immediately after $\overline{\text{ERROR}}$ pin reset request is received.

Figure 22-19. $\overline{\text{ERROR}}$ Pin Timing - Example 3

ADVANCE INFORMATION

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.

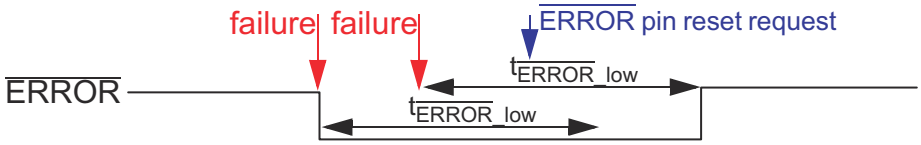


Figure 22-20. $\overline{\text{ERROR}}$ Pin Timing - Example 4

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.

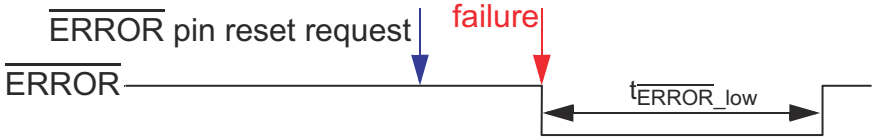


Figure 22-21. $\overline{\text{ERROR}}$ Pin Timing - Example 5

ADVANCE INFORMATION

22.4.4.3 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.
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.

22.4.5 Recommended Programming Procedure

During the initialization stage, the application code should follow the recommendations in Figure 22-22 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 $\overline{\text{RST}}$.

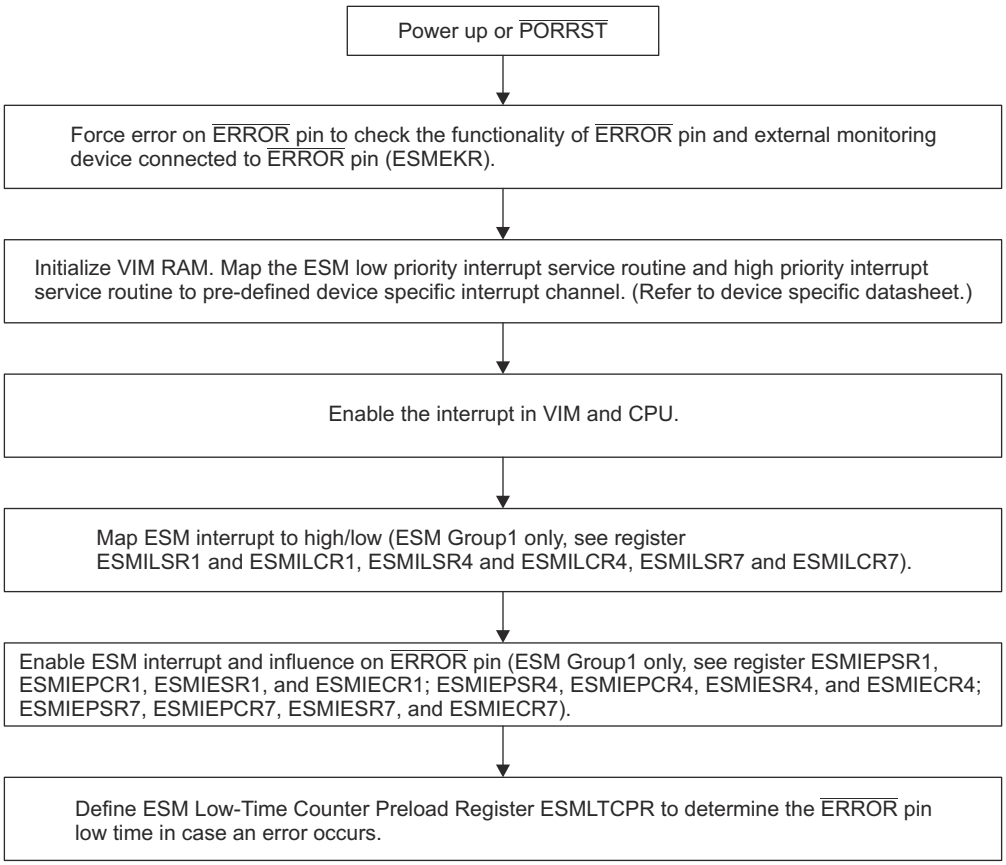


Figure 22-22. ESM Initialization

22.4.6 Integration Details

LPR has one instance of the ESM module

	Parameters		
Instance	Max Group 1	Max Group 2	Max Group 3
APPSS_ESM	128	32	32

The ESM output port is connected to the PAD.

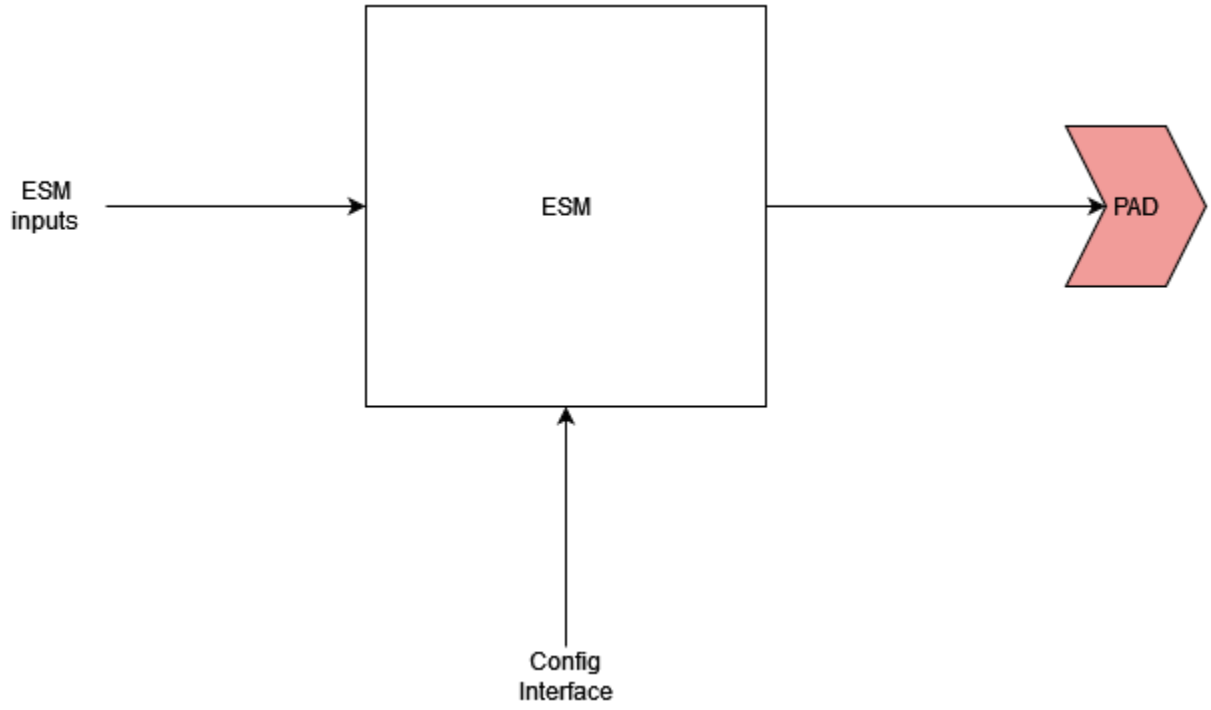


Figure 22-23. ESM Inputs

The ESM error output is connected to the PAD directly. There is no polarity inversion at the top-level.

22.4.7 External event ESM masking

The ESM Instance has the following external masking logic on the Group Events.

	External Masking Logic		
Instance	Group 1	Group 2	Group 3
APPSS_ESM	None	Refer APP_CTRL:ESM_GATINGx	Refer APP_CTRL:ESM_GATINGx

The high priority and low priority interrupts from ESM go to APPSS interrupts. For details, refer to **Section 12.2. APPSS Interrupt Map.**

The details of the interrupts going to ESM is captured in the APP_SS ESM Mapping below.

Table 22-87. APP_SS ESM Mapping

ESM GROUP1	Define Name	Description	Comments	Subsystem	Module	Port Name
31	ACESS_ERROR	Aggregated address access error.				
30	TOPSS_AGG_ERR	Aggregated TOPSS Errors - FRC undefined state error				
29	PLL_DIG_LOC_MON	Lock monitor signal from the PLL_DIG				
28	PM_AGG_ERR	Aggregated Error Signals from PM. RCOSC10M_GOOD , RCOSC32K_GOOD, SUPPLY_OK etc				
27	ANA_AGG_ERR	Aggregated Error Signals from Analog- Saturation detection, APLL Lock Monitor				
26	EFUSE_ERR	EFUSE Error				
25	QSPI_WR_ERR	QSPI write error				

Table 22-87. APP_SS ESM Mapping (continued)

ESM GROUP1	Define Name	Description	Comments	Subsystem	Module	Port Name
24	MPU_PROT_AGG_ERR	Aggregated MPU Protection Error for MPUs				
23	APPSS_MCRC_ERR	MCRC Comparision Error				
22	APPSS_DCC_ERR	DCC frequency comparision error				
21	APPSS_MCAN_AGG_ERR	Aggregated MCAN Errors - MCAN_SERR - Single Bit correctable error indication for MCANA Message Memory - MCAN_UERR- Multi Bit uncorrectable error indication for MCANA Message Memory - MCAN_TS_ERR - MCANA Timestamping Error				
20	APPSS_TPCC_AGG_ERR	APPSS_TPCC Aggregated Error Interrupt - TPCC Error - TPTC Error for all TPTCs connected to TPCC - Read and Write Config Space Access error to TPCC - Read and Write Config Space Access error or all TPTCs connected to TPCC				
19	HWASS_TPCC_AGG_ERR	HWASS_TPCC Aggregated Error Interrupt - TPCC Error - TPTC Error for all TPTCs connected to TPCC - Read and Write Config Space Access error to TPCC - Read and Write Config Space Access error or all TPTCs connected to TPCC				
18	CM3_LBIST_ERR	LBIST Error indication for Cortex M3				
17	CM4F_LBIST_ERR	LBIST Error indication for Cortex M4F				
16	HWA_AGG_ERR	Aggregated HWA Errors - iping/ipong/oping/opong memories access error - FSM Lock step error - window ram parity error - iping/ipong/oping/opong memories parity error - shared memory invalid address access error				
15	HWA_LOCAL_RAM_ECC_AGG_SERR	Aggregated ECC multi-bit error from HWA local RAMs - HWA param ram - ADCBUF ping/pong memories - HWASS TPTC1 FIFO - HWASS TPTC2 FIFO				
14	HWA_LOCAL_RAM_ECC_AGG_UERR	Aggregated ECC single-bit error from HWA local RAMs - HWA param ram - ADCBUF ping/pong memories - HWASS_TPTC1 FIFO - HWASS_TPTC2 FIFO				
13	HWASS_SHARED_RAM_ECC_AGG_SERR	ECC single-bit error indication from shared memory banks owned by HWASS				

Table 22-87. APP_SS ESM Mapping (continued)

ESM GROUP1	Define Name	Description	Comments	Subsystem	Module	Port Name
12	HWASS_SHARED_RAM_ECC_AGG_UERR	ECC multi-bit (uncorrectable) error indication from shared memory banks owned by HWASS				
11	FECSS_AGG_ERR	Aggregated Error from DFE, Timing Engines and other FECSS modules - Parity error from timing engine - undefined_st_entered error from timing engine				
10	FECSS_ECC_AGG_SERR	Aggregated ECC single-bit error (from other FECSS RAMs than CPU RAMs) - GPADC ram				
9	FECSS_ECC_AGG_UERR	Aggregated ECC multi-bit error (from other FECSS RAMs than CPU RAMs) -GPADC ram				
8	FECSS_CM3_RAM_ECC_AGG_SERR	Aggregated ECC single-bit error from FECSS CPU RAMs and 96 KB shared ram when it is shared				
7	FECSS_CM3_RAM_ECC_AGG_UERR	Aggregated ECC multi-bit error from FECSS CPU RAMs and 96 KB shared ram when it is shared				
6	APPSS_ECC_AGG_SERR	Aggregated ECC single-bit error from APPSS RAMs (other than CPU RAMs) - APPSS_TPTC1 FIFO - APPSS_TPTC2 FIFO				
5	APPSS_ECC_AGG_UERR	Aggregated ECC multi-bit (Uncorrectbale) error from APPSS RAMs (other than CPU RAMs) - APPSS_TPTC1 FIFO - APPSS_TPTC2 FIFO				
4	APPSS_CM4F_RAM_ECC_AGG_SERR	Aggregated ECC single-bit error from APPSS Cortex M4F ROM, RAMs and 128KB/256 KB shared ram when it is shared				
3	APPSS_CM4F_RAM_ECC_AGG_UERR	Aggregated ECC multi-bit (Uncorrectable) error from APPSS Cortex M4F ROM, RAMs and 128KB/256 KB shared ram when it is shared				
2	APPSS_WDT_NMI	APPSS Watch dog timer non maskable irq				
1	HSM_ESM_LO	RESERVED for future scalability				
0	HSM_ESM_HI	RESERVED for future scalability				
ESM GROUP3	**Define Name**	**Description**	**Comments**	**Subsystem**	**Module**	**Port Name**
7 to 4	RESERVED	RESERVED				
3	CLKM_LIMP_MODE	from PRCM. Error for Reference clock not ticking.				

Table 22-87. APP_SS ESM Mapping (continued)

ESM GROUP1	Define Name	Description	Comments	Subsystem	Module	Port Name
2	APPSS_CM4F_RAM_ECC_AGG_UERR	Aggregated ECC multi-bit (Uncorrectable) error from APPSS Cortex M4F ROM, RAMs and 128KB/256 KB shared ram when it is shared				
1	EFUSE_AUTOLOAD_ERR	Efuse autoload error				
0	RESERVED	RESERVED				

22.4.8 ESM Group1

22.4.9 ESM Group 2

Not supported.

22.4.10 ESM Group 3

ESM GROUP3	Define Name	Description
7 to 4	RESERVED	RESERVED
3	CLKM_LIMP_MODE	from PRCM. Error for Reference clock not ticking.
2	APPSS_CM4F_RAM_ECC_AGG_UERR	Aggregated ECC multi-bit (Uncorrectable) error from APPSS Cortex M4F ROM, RAMs and 128KB/256 KB shared ram when it is shared
1	EFUSE_AUTOLOAD_ERR	Efuse autoload error
0	RESERVED	RESERVED

TPCC Error Aggregation Scheme:

The following interrupts are aggregated and sent to the ESM

- TPCC Error
- TPCC MPU Error
- TPTCs Error
- TPCC Read and Write Config Space Access error
- TPTCs Read and Write Config Space Access error

TPCC	ESM Interrupt
APPSS_TPCC_A	APPSS_TPCC_AGG_ERR
APPSS_TPCC_B (HWASS_TPCC)	HWASS_TPCC_AGG_ERR

For an event to generate an interrupt to the processor, the corresponding bit field must be unmasked in TPCC_x_ERRAGG_MASK.

On an interrupt processor can read the TPCC_x_ERRAGG_STATUS register to detect which event triggered the interrupt.

The interrupt can be cleared by writing 0x1 to the corresponding bit in TPCC_x_ERRAGG_STATUS.

It is the SW responsibility to ensure that all the aggregated interrupts are cleared so that the level interrupt is deserted before exiting the ISR. Only then it is guaranteed that a new pulse interrupt will be generated to the processor. Hence after Clearing SW should read the register to confirm a value of 0x0.

The register TPCC_x_ERRAGG_STATUS_RAW is set on an event irrespective of the value in TPCC_x_ERRAGG_MASK. This field can be cleared by writing 0x1 to the corresponding bit in TPCC_x_ERRAGG_STATUS_RAW.

Other Aggregator Errors

30	TOPSS_AGG_ERR	Aggregated TOPSS Errors - FRC undefined state error
28	PM_AGG_ERR	Aggregated Error Signals from PM. RCOSC10M_GOOD , RCOSC32K_GOOD, SUPPLY_OK.
27	ANA_AGG_ERR	Aggregated Error Signals from Analog-Saturation detection, APLL Lock Monitor
24	MPU_PROT_AGG_ERR	Aggregated MPU Protection Error for MPUs
21	APPSS_MCAN_AGG_ERR	Aggregated MCAN Errors - MCAN_SERR - Single Bit correctable error indication for MCANA Message Memory - MCAN_UERR- Multi Bit uncorrectable error indication for MCANA Message Memory - MCAN_TS_ERR - MCANA Timestamping Error
16	HWA_AGG_ERR	Aggregated HWA Errors - iping/ipong/oping/ opong memories access error - FSM Lock step error - window ram parity error - iping/ ipong/oping/opong memories parity error - shared memory invalid address access error
11	FECSS_AGG_ERR	Aggregated Error from DFE, Timing Engines and other FECSS modules - Parity error from timing engine - undefined_st_entered error from timing engine

Error Scenarios and Corresponding Error Aggregation:

SCENARIO 1 : ACCESS Error from Shared Memory . Error is generated when shared-memory region is wrongly accessed.

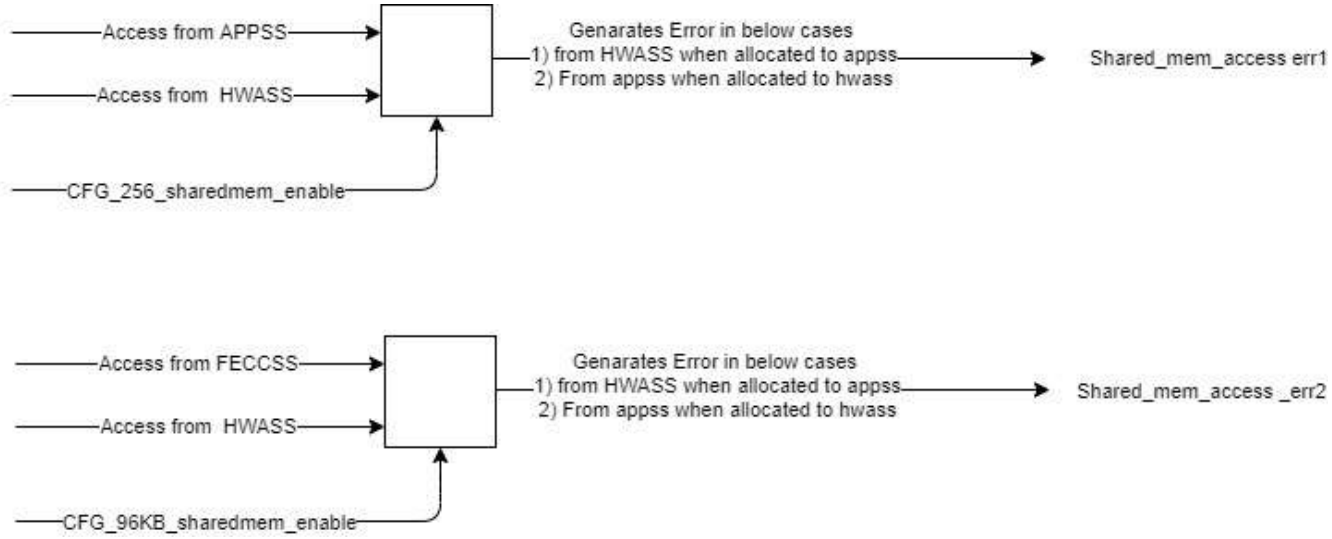


Figure 22-24. ACCESS Error from Shared Memory

SCENARIO 2: Access Errors (both write and read) from control-spaces and MPUs mentioned below. Error is generated when unallocated location is accessed in CTRL-Spaces and MPU.

Aggregations are performed in individual subsystems and lines are merged and provided to ESM line 31.

FECSS: Taken care through DFP

APPSS:

- APP_CTRL
- APP_RCM

- APP_MPU
- APP_IDALLOC
- APP_AHB
- APP_SHARED_ACC_ERR

HWASS:

- ADCBUF_CTRL

TOPSS:

- PLL_DIG_CTRL
- TOPSS_CTRL
- APLL_CTRL
- FRC (Frame Timer)
- TOP_EFUSE (TOP_CTRL)
- TOP_PRCM

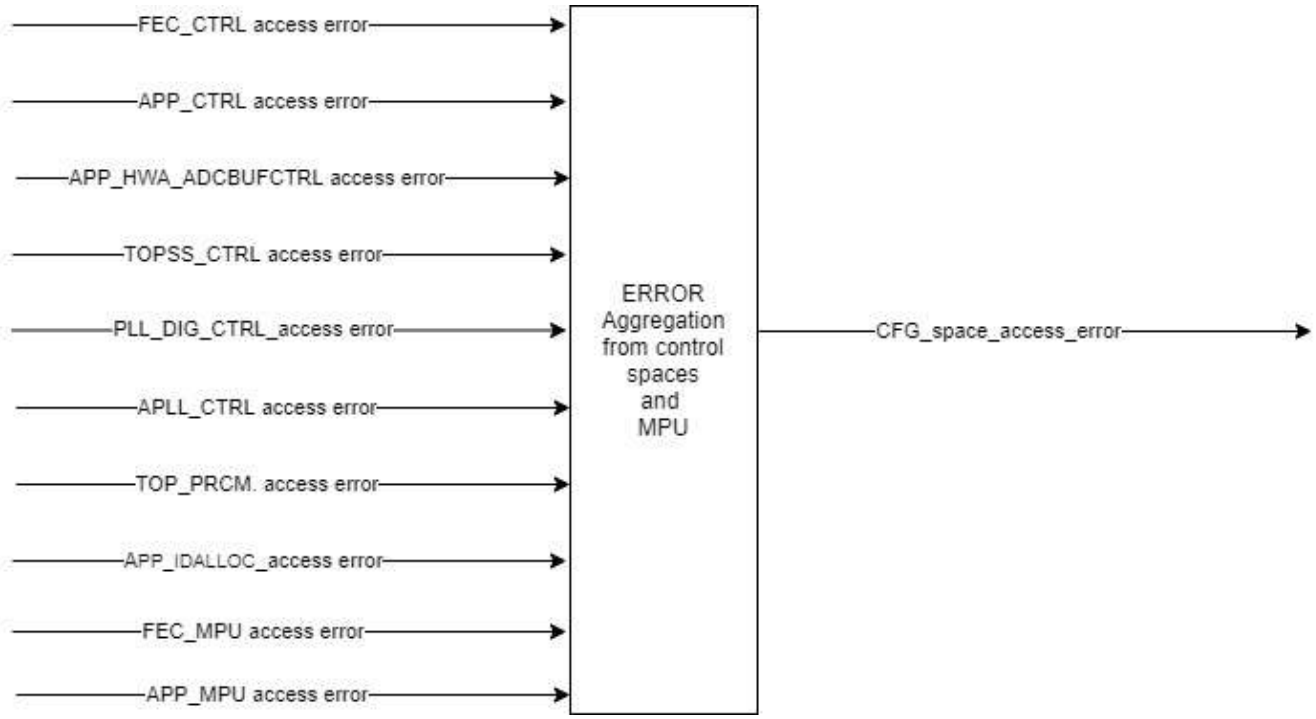


Figure 22-25. Access Error-CFG_space_access_errr

SCENARIO 3: AHB Write response ERRORs. AHB2VBUSP bridge does not use write-response from protocol. So write-response from VBUSP protocol is used in parallel to generate ERROR.

Figure 22-26. Access Error-AHB Write Response Error

SCENARIO 4: Accessing a power-domain when it is switched off should generate a expectation to processor. Isolation values of all the below mentioned signals are tied high. So both read/write-accesses to these power-domains wont hang and read-access generated abort to the processor.

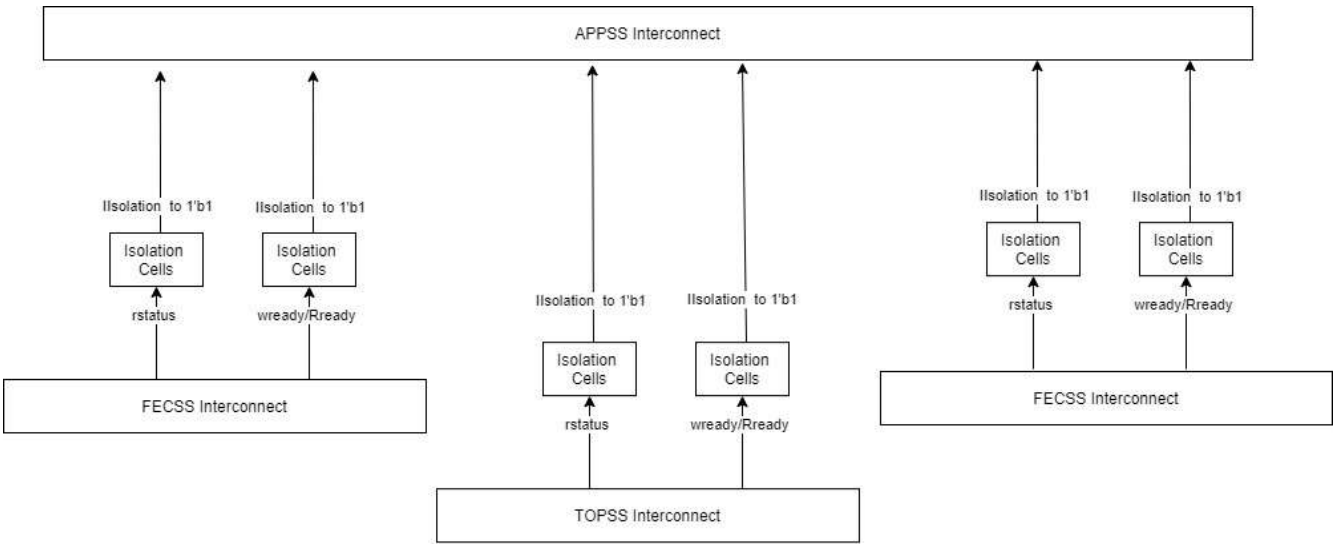


Figure 22-27. Access Error-Power Domain Accesses

Scenario	M3	M4	HWA TPTC	APP TPTC
96 KB Shared Mem (Shared with HWA) – 0x2120xxxx Or Powered Down	No ESM is generated	No ESM and No Bus fault No ESM is generated	No ESM No ESM is generated	No ESM No ESM is generated
256 KB Shared Mem (Shared with HWA) – 0x2248xxxx Or Powered Down	Rd,WR – ESM (ESM-31) RD-> No busfault	RD-> no esm & bus-fault	RD,WR- ESM (ESM-31 from fecss) Tpcc err int	RD,WR- ESM (ESM-31 from fecss) Tpcc err int
96 KB Shared Mem (Shared with M3) – 0x6000xxxx Or Powered Down	ESM HWA_AGG_ERR (ESM16)	ESM HWA_AGG_ERR (ESM16)	ESM HWA_AGG_ERR (ESM16)	ESM HWA_AGG_ERR (ESM16)
256 KB Shared Mem (Shared with M4) – 0x6000xxxx Or Powered Down	ESM HWA_AGG_ERR (ESM16)	ESM HWA_AGG_ERR (ESM16)	ESM HWA_AGG_ERR (ESM16)	ESM HWA_AGG_ERR (ESM16)
Invalid address range access to CM4 ROM/ RAM1/RAM2/RAM3	Wr : ESM Rd : no fault or no ESM	Wr : Bus fault Rd : Bus Fault	Wr : ESM Rd : no fault or no ESM	Wr : ESM Rd : no fault or no ESM
Invalid address range access to CM3ROM/ RAM1	Wr : Bus fault Rd : Bus Fault	Wr : ESM Rd : no fault or no ESM	Wr : ESM Rd : no fault or no ESM	Wr : ESM Rd : no fault or no ESM

SCENARIO5: ACCESS error due memory in power down state

Mask register : APP_CTRL Registers:APPSS_ERRAGG_MASK1

Status register : APP_CTRL Registers:APPSS_ERRAGG_STATUS1

When memory is on power down state i.e. AON and AGOOD are low then either Bus Fault will be generated or the separate cluster esm error will be generated. (In some cases both errors will be generated)

Table 22-88. Error Reporting for Accessing Invalid Address Range

Sr. No.	Cluster	Memory	Write	Read
			Cluster ESM error	Cluster ESM error
1	Cluser 1		Supported	Supported

Table 22-88. Error Reporting for Accessing Invalid Address Range (continued)

Sr. No.	Cluster	Memory	Write	Read
2	Cluser 2	CPU RAM	Supported	Supported
		Ram2KB	Supported	Supported
3	Cluser 3		Supported	Supported
4	Cluser 4		Supported	Supported
5	Cluser 5		Supported	Supported
6	Cluser 6		Supported	Supported
7	Cluser 7	16KB bank 1	Supported	Supported
		Timing Ram	Supported	Supported
8	Cluser 8	16KB bank 2	Supported	Supported
		BISTFFT, GPADC	Supported	Supported
9	Cluser 9	96KB shared ram	Supported	Supported
10	Cluser 10	Param Ram	Supported	Supported
11	Cluser 11		Supported	Supported
12	Cluser 12	160KB Shared ram	Supported	Supported

22.4.11 APP_ESM Registers

Table 22-89 lists the memory-mapped registers for the APP_ESM registers. 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. APP_ESM Registers

Offset	Acronym	Register Name	Section
0h	ESMIEPSR1	ESM Enable ERROR Pin Action/Response Register 1	Go
4h	ESMIEPCR1	ESM Disable ERROR Pin Action/Response Register 1	Go
8h	ESMIESR1	ESM Interrupt Enable Set/Status Register 1	Go
Ch	ESMIECR1	ESM Interrupt Enable Clear/Status Register 1	Go
10h	ESMILSR1	Interrupt Level Set/Status Register 1	Go
14h	ESMILCR1	Interrupt Level Clear/Status Register 1	Go
18h	ESMSR1	ESM Status Register 1	Go
1Ch	ESMSR2	ESM Status Register 2	Go
20h	ESMSR3	ESM Status Register 3	Go
24h	ESMEPSR	ESM ERROR Pin Status Register	Go
28h	ESMIOFFHR	ESM Interrupt Offset High Register	Go
2Ch	ESMIOFFLR	ESM Interrupt Offset Low Register	Go
30h	ESMLTCR	ESM Low-Time Counter Register	Go
34h	ESMLTCPR	ESM Low-Time Counter Preload Register	Go
38h	ESMEKR	ESM Error Key Register	Go
3Ch	ESMSSR2	ESM Status Shadow Register 2	Go
40h	ESMIEPSR4	ESM Enable ERROR Pin Action/Response Register 4	Go
44h	ESMIEPCR4	ESM Disable ERROR Pin Action/Response Register 4	Go
48h	ESMIESR4	ESM Interrupt Enable Set/Status Register 4	Go
4Ch	ESMIECR4	ESM Interrupt Enable Clear/Status Register 4	Go
50h	ESMILSR4	Interrupt Level Set/Status Register 4	Go
54h	ESMILCR4	Interrupt Level Clear/Status Register 4	Go
58h	ESMSR4	ESM Status Register 4	Go
80h	ESMIEPSR7	ESM Enable ERROR Pin Action/Response Register 7	Go
84h	ESMIEPCR7	ESM Disable ERROR Pin Action/Response Register 7	Go
88h	ESMIESR7	ESM Interrupt Enable Set/Status Register 7	Go
8Ch	ESMIECR7	ESM Interrupt Enable Clear/Status Register 7	Go
90h	ESMILSR7	Interrupt Level Set/Status Register 7	Go
94h	ESMILCR7	Interrupt Level Clear/Status Register 7	Go
98h	ESMSR7	ESM Status Register 7	Go
C0h	ESMIEPSR10	ESM Enable ERROR Pin Action/Response Register 10	Go
C4h	ESMIEPCR10	ESM Disable ERROR Pin Action/Response Register 10	Go
C8h	ESMIESR10	ESM Interrupt Enable Set/Status Register 10	Go
CCh	ESMIECR10	ESM Interrupt Enable Clear/Status Register 10	Go
D0h	ESMILSR10	Interrupt Level Set/Status Register 10	Go
D4h	ESMILCR10	Interrupt Level Clear/Status Register 10	Go
D8h	ESMSR10	ESM Status Register 10	Go

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. APP_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

22.4.11.1 ESMIEPSR1 Register (Offset = 0h) [Reset = 00000000h]

ESMIEPSR1 is shown in Table 22-91.

Return to the Summary Table.

ESM Enable ERROR Pin Action/Response Register 1

Table 22-91. 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.

22.4.11.2 ESMIEPCR1 Register (Offset = 4h) [Reset = 00000000h]

ESMIEPCR1 is shown in Table 22-92.

Return to the Summary Table.

ESM Disable ERROR Pin Action/Response Register 1

Table 22-92. 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.

22.4.11.3 ESMIESR1 Register (Offset = 8h) [Reset = 00000000h]

ESMIESR1 is shown in Table 22-93.

Return to the Summary Table.

ESM Interrupt Enable Set/Status Register 1

Table 22-93. 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.

22.4.11.4 ESMIECR1 Register (Offset = Ch) [Reset = 00000000h]

ESMIECR1 is shown in Table 22-94.

Return to the Summary Table.

ESM Interrupt Enable Clear/Status Register 1

Table 22-94. 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.

22.4.11.5 ESMILSR1 Register (Offset = 10h) [Reset = 00000000h]

ESMILSR1 is shown in Table 22-95.

Return to the Summary Table.

Interrupt Level Set/Status Register 1

Table 22-95. 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.

22.4.11.6 ESMILCR1 Register (Offset = 14h) [Reset = 00000000h]

ESMILCR1 is shown in Table 22-96.

Return to the Summary Table.

Interrupt Level Clear/Status Register 1

Table 22-96. 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.

22.4.11.7 ESMSR1 Register (Offset = 18h) [Reset = 00000000h]

ESMSR1 is shown in Table 22-97.

Return to the Summary Table.

ESM Status Register 1

Table 22-97. 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.

22.4.11.8 ESMSR2 Register (Offset = 1Ch) [Reset = 00000000h]

ESMSR2 is shown in Table 22-98.

Return to the Summary Table.

ESM Status Register 2

Table 22-98. 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.

22.4.11.9 ESMSR3 Register (Offset = 20h) [Reset = 00000000h]

ESMSR3 is shown in Table 22-99.

Return to the Summary Table.

ESM Status Register 3

Table 22-99. 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.

22.4.11.10 ESMEPSR Register (Offset = 24h) [Reset = 00000000h]

ESMEPSR is shown in Table 22-100.

Return to the Summary Table.

ESM ERROR Pin Status Register

Table 22-100. 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.

22.4.11.11 ESMIOFFHR Register (Offset = 28h) [Reset = 00000000h]

ESMIOFFHR is shown in Table 22-101.

Return to the Summary Table.

ESM Interrupt Offset High Register

Table 22-101. ESMIOFFHR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	Read returns 0. Writes have no effect.

Table 22-101. ESMIOFFHR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8-0	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.

22.4.11.12 ESMIOFFLR Register (Offset = 2Ch) [Reset = 00000000h]

ESMIOFFLR is shown in Table 22-102.

Return to the Summary Table.

ESM Interrupt Offset Low Register

Table 22-102. ESMIOFFLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
7-0	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.

22.4.11.13 ESMLTCR Register (Offset = 30h) [Reset = 00003FFFh]

ESMLTCR is shown in Table 22-103.

Return to the Summary Table.

ESM Low-Time Counter Register

Table 22-103. 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	3FFFh	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

22.4.11.14 ESMLTCPR Register (Offset = 34h) [Reset = 00003FFFh]

ESMLTCPR is shown in Table 22-104.

Return to the Summary Table.

ESM Low-Time Counter Preload Register

Table 22-104. 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	3FFFh	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).

22.4.11.15 ESMEKR Register (Offset = 38h) [Reset = 00000000h]

ESMEKR is shown in Table 22-105.

Return to the Summary Table.

ESM Error Key Register

Table 22-105. 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.

22.4.11.16 ESMSSR2 Register (Offset = 3Ch) [Reset = 00000000h]

ESMSSR2 is shown in Table 22-106.

Return to the Summary Table.

ESM Status Shadow Register 2

Table 22-106. 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).

22.4.11.17 ESMIEPSR4 Register (Offset = 40h) [Reset = 00000000h]

ESMIEPSR4 is shown in Table 22-107.

Return to the Summary Table.

ESM Enable ERROR Pin Action/Response Register 4

Table 22-107. 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.

22.4.11.18 ESMIEPCR4 Register (Offset = 44h) [Reset = 00000000h]

ESMIEPCR4 is shown in Table 22-108.

Return to the Summary Table.

ESM Disable ERROR Pin Action/Response Register 4

Table 22-108. 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.

22.4.11.19 ESMIESR4 Register (Offset = 48h) [Reset = 00000000h]

ESMIESR4 is shown in Table 22-109.

Return to the Summary Table.

ESM Interrupt Enable Set/Status Register 4

Table 22-109. 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.

22.4.11.20 ESMIECR4 Register (Offset = 4Ch) [Reset = 00000000h]

ESMIECR4 is shown in Table 22-110.

Return to the Summary Table.

ESM Interrupt Enable Clear/Status Register 4

Table 22-110. 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.

22.4.11.21 ESMILSR4 Register (Offset = 50h) [Reset = 00000000h]

ESMILSR4 is shown in Table 22-111.

Return to the Summary Table.

Interrupt Level Set/Status Register 4

Table 22-111. 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.

22.4.11.22 ESMILCR4 Register (Offset = 54h) [Reset = 00000000h]

ESMILCR4 is shown in Table 22-112.

Return to the Summary Table.

Interrupt Level Clear/Status Register 4

Table 22-112. 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.

22.4.11.23 ESMSR4 Register (Offset = 58h) [Reset = 00000000h]

ESMSR4 is shown in Table 22-113.

Return to the Summary Table.

ESM Status Register 4

Table 22-113. 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.

22.4.11.24 ESMIEPSR7 Register (Offset = 80h) [Reset = 00000000h]

ESMIEPSR7 is shown in Table 22-114.

Return to the Summary Table.

ESM Enable ERROR Pin Action/Response Register 7

Table 22-114. 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.

22.4.11.25 ESMIEPCR7 Register (Offset = 84h) [Reset = 00000000h]

ESMIEPCR7 is shown in Table 22-115.

Return to the Summary Table.

ESM Disable ERROR Pin Action/Response Register 7

Table 22-115. 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.

22.4.11.26 ESMIESR7 Register (Offset = 88h) [Reset = 00000000h]

ESMIESR7 is shown in Table 22-116.

Return to the Summary Table.

ESM Interrupt Enable Set/Status Register 7

Table 22-116. 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.

22.4.11.27 ESMIECR7 Register (Offset = 8Ch) [Reset = 00000000h]

ESMIECR7 is shown in Table 22-117.

Return to the Summary Table.

ESM Interrupt Enable Clear/Status Register 7

Table 22-117. 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.

22.4.11.28 ESMILSR7 Register (Offset = 90h) [Reset = 00000000h]

ESMILSR7 is shown in Table 22-118.

Return to the Summary Table.

Interrupt Level Set/Status Register 7

Table 22-118. 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.

22.4.11.29 ESMILCR7 Register (Offset = 94h) [Reset = 00000000h]

ESMILCR7 is shown in Table 22-119.

Return to the Summary Table.

Interrupt Level Clear/Status Register 7

Table 22-119. 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.

22.4.11.30 ESMSR7 Register (Offset = 98h) [Reset = 00000000h]

ESMSR7 is shown in Table 22-120.

Return to the Summary Table.

ESM Status Register 7

Table 22-120. 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.

22.4.11.31 ESMIEPSR10 Register (Offset = C0h) [Reset = 00000000h]

ESMIEPSR10 is shown in Table 22-121.

Return to the Summary Table.

ESM Enable ERROR Pin Action/Response Register 10

Table 22-121. ESMIEPSR10 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 ESMIEPCR10 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 ESMIEPCR10 register.

22.4.11.32 ESMIEPCR10 Register (Offset = C4h) [Reset = 00000000h]

ESMIEPCR10 is shown in Table 22-122.

Return to the Summary Table.

ESM Disable ERROR Pin Action/Response Register 10

Table 22-122. ESMIEPCR10 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 ESMIEPSR10 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 ESMIEPSR10 register.

22.4.11.33 ESMIESR10 Register (Offset = C8h) [Reset = 00000000h]

ESMIESR10 is shown in Table 22-123.

Return to the Summary Table.

ESM Interrupt Enable Set/Status Register 10

Table 22-123. ESMIESR10 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 ESMIECR10 register unchanged. 1 Read: Interrupt is enabled. Write: Enables interrupt and sets the corresponding clear bit in the ESMIECR10 register.

22.4.11.34 ESMIECR10 Register (Offset = CCh) [Reset = 00000000h]

ESMIECR10 is shown in Table 22-124.

Return to the Summary Table.

ESM Interrupt Enable Clear/Status Register 10

Table 22-124. ESMIECR10 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 ESMIESR10 register unchanged. 1 Read: Interrupt is enabled. Write: Disables interrupt and clears the corresponding set bit in the ESMIESR10 register.

22.4.11.35 ESMILSR10 Register (Offset = D0h) [Reset = 00000000h]

ESMILSR10 is shown in Table 22-125.

Return to the Summary Table.

Interrupt Level Set/Status Register 10

Table 22-125. ESMILSR10 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 ESMILCR10 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 ESMILCR10 register.

22.4.11.36 ESMILCR10 Register (Offset = D4h) [Reset = 00000000h]

ESMILCR10 is shown in Table 22-126.

Return to the Summary Table.

Interrupt Level Clear/Status Register 10

Table 22-126. ESMILCR10 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 ESMILSR10 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 ESMILSR10 register.

22.4.11.37 ESMSR10 Register (Offset = D8h) [Reset = 00000000h]

ESMSR10 is shown in Table 22-127.

Return to the Summary Table.

ESM Status Register 10

Table 22-127. ESMSR10 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.

22.5 Cyclic Redudancy Check (CRC)

This section describes the cyclic redundancy check (CRC) controller module. Presently one CRC modules have been instantiated in the device for APPSS.

22.5.1 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.

22.5.2 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.

22.5.3 Block Diagram

Figure 22-28 shows a block diagram of the CRC controller.

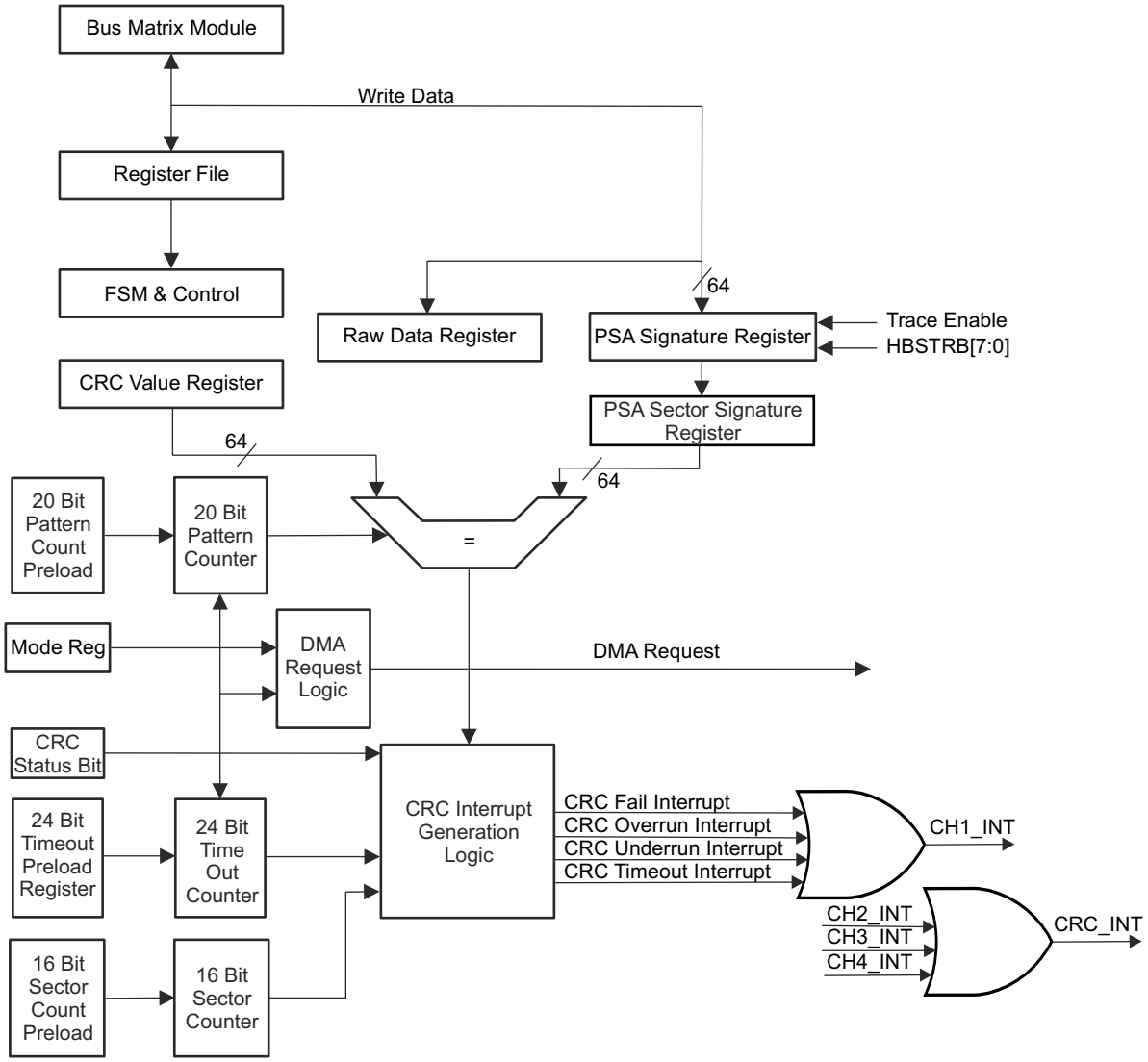


Figure 22-28. CRC Controller Block Diagram For One Channel

22.5.4 Module Operation

22.5.4.1 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.

22.5.4.2 CRC Modes of Operation

CRC Controller can operate in AUTO, Semi-CPU, and Full-CPU modes.

22.5.4.2.1 AUTO Mode

In AUTO mode, CRC Controller in conjunction with DMA controller can perform CRC 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.

22.5.4.2.2 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 compares it to a pre-determined CRC value.

22.5.4.2.3 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.

22.5.4.3 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 22-29.

$$f(x) = x^{64} + x^4 + x^3 + x + 1 \quad (10)$$

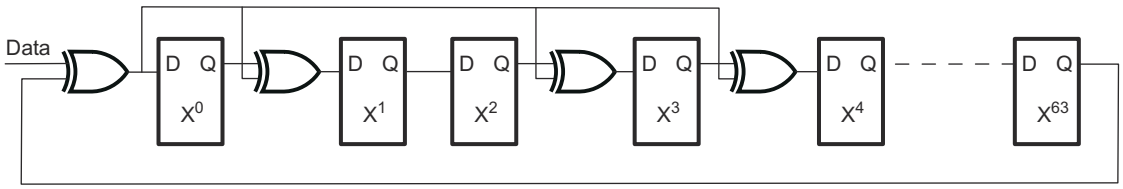


Figure 22-29. Linear Feedback Shift Register (LFSR)

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 controller. 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

22.5.4.4 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.

22.5.4.5 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.

22.5.4.6 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.

22.5.4.7 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.

22.5.4.7.1 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 22-30 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 22-30, an HW DMA request from a timer is used as a trigger source to initiate DMA transfer. If all two CRC channels are active in AUTO mode then a total of two DMA requests would be generated by CRC Controller.

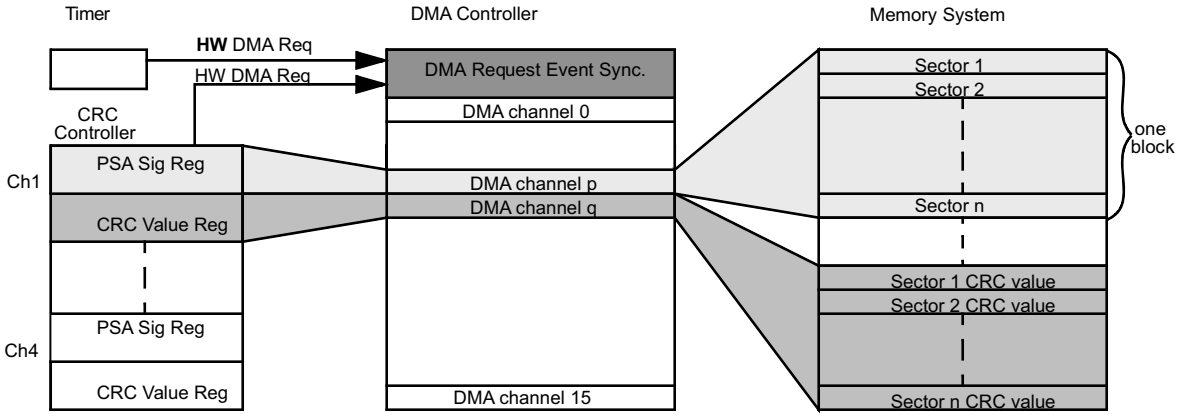


Figure 22-30. AUTO Mode Using Hardware Timer Trigger

22.5.4.7.2 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 22-31 for an illustration.

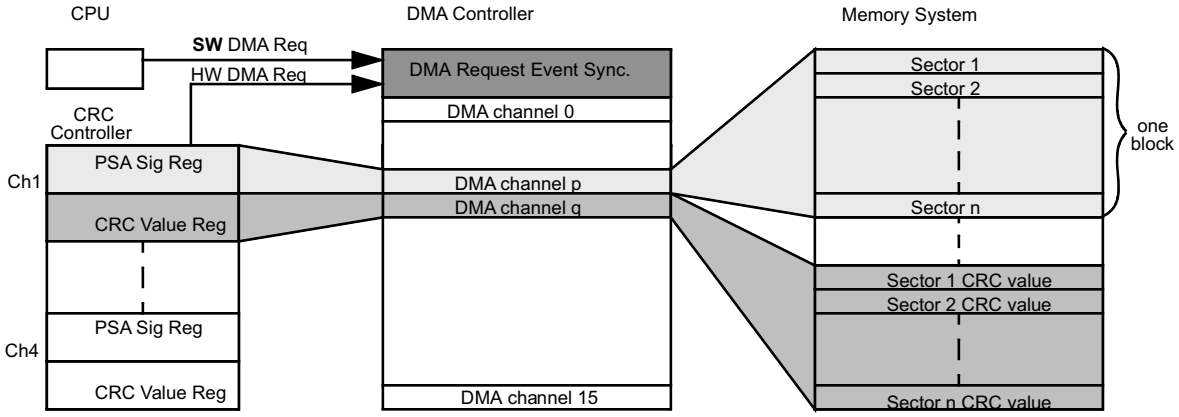


Figure 22-31. AUTO Mode With Software CPU Trigger

ADVANCE INFORMATION

22.5.4.7.3 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 22-32 illustrates the DMA setup using semi-CPU mode with hardware timer trigger.

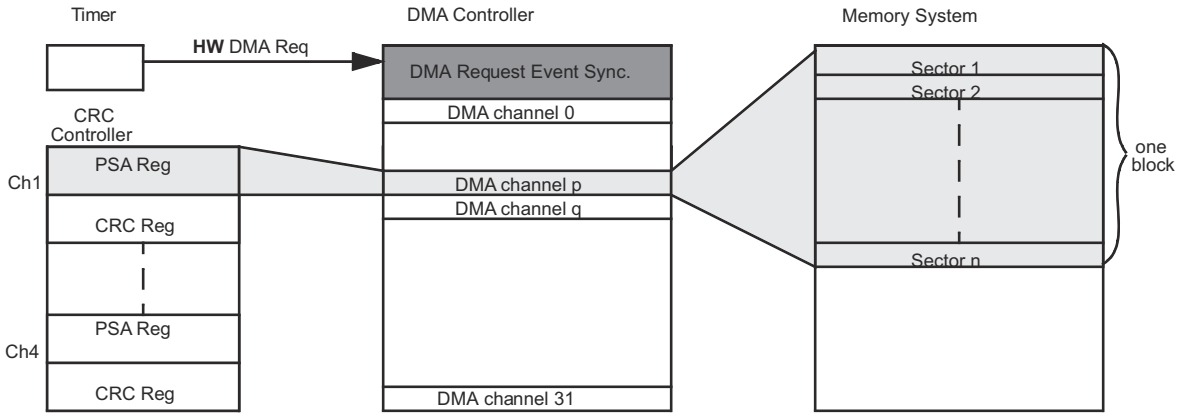


Figure 22-32. Semi-CPU Mode With Hardware Timer Trigger

Table 22-128. 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

22.5.4.8 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.

22.5.4.9 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.

22.5.4.10 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 22-129. 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

22.5.4.10.1 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.

22.5.4.10.2 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.

22.5.4.10.3 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.

22.5.4.10.4 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.

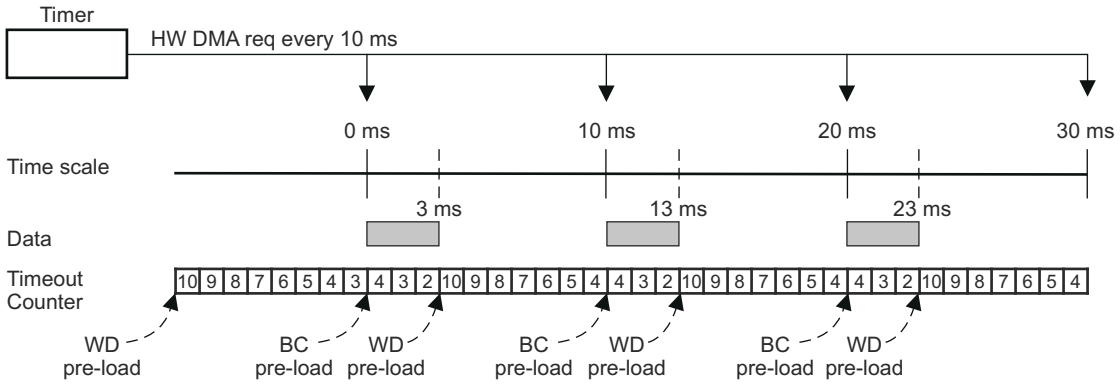
22.5.4.10.5 Timeout Interrupt

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.

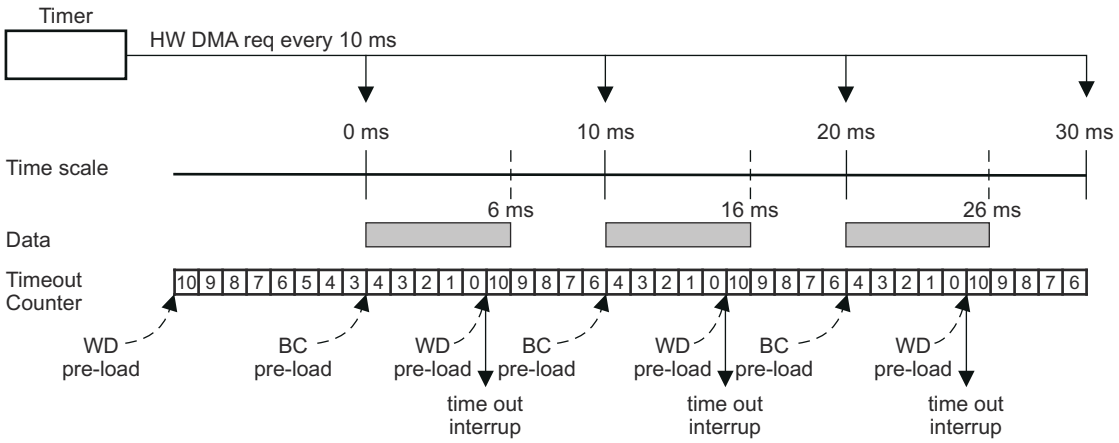
In Figure 22-33, a timer generates DMA request every 10ms to trigger one block (pattern count x sector count) transfer. Since we want to make sure that DMA does start to transfer a block every 10 ms we would set the first pre-load value to 10ms in CRC_WDTOPLDx. We also want to make sure that one block 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.



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.

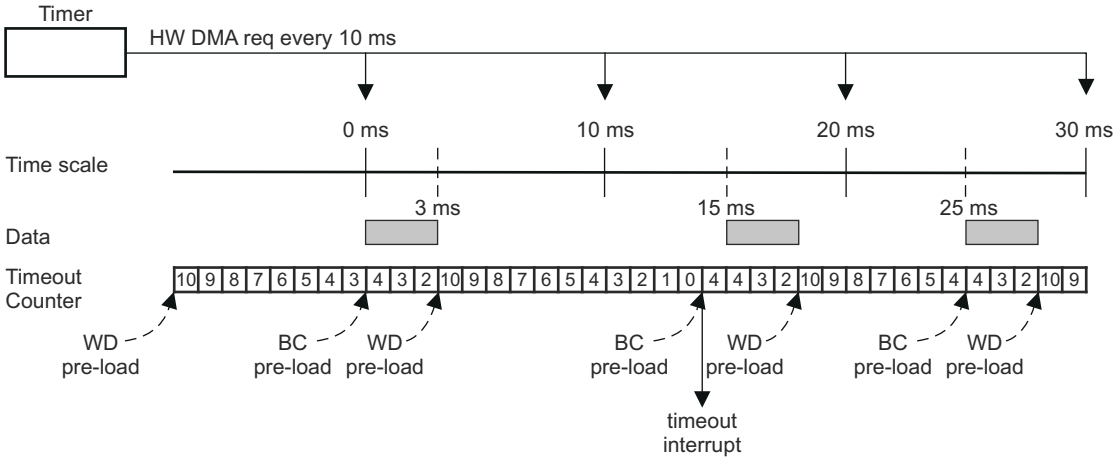
Figure 22-33. Timeout Example 1



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 22-34. 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 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.

Figure 22-35. Timeout Example 3

22.5.4.10.6 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 22-130 shows the offset interrupt vector address of each interrupt condition in an ascending order of priority.

Table 22-130. 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

22.5.4.10.7 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.

22.5.4.11 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.

22.5.4.12 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 Controller bus error should be generated if reading from the unimplemented locations.

22.5.4.13 Peripheral Bus Interface

CRC is a Peripheral target 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.

22.5.5 Example

This section illustrates several of the ways in which the CRC Controller can be utilized to perform CRC.

22.5.5.1 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.

22.5.5.1.1 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.

22.5.5.1.2 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.

22.5.5.1.3 CRC Setup

- Program the pattern count to 128.
- Program the sector count to 2048.
- 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.

22.5.5.2 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.

22.5.5.2.1 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.

22.5.5.2.2 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.

22.5.5.3 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.

22.5.5.3.1 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.

22.5.5.3.2 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.

22.5.5.3.3 CRC Setup

- Program the pattern count to 128.
- Program the sector count to 2048.
- 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.

22.5.5.4 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.

22.5.5.4.1 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.

22.5.6 APP_CRC Registers

Table 22-131 lists the memory-mapped registers for the APP_CRC registers. All register offset addresses not listed in Table 22-131 should be considered as reserved locations and the register contents should not be modified.

Table 22-131. APP_CRC Registers

Offset	Acronym	Register Name	Section
0h	CRC_CTRL0	CRC Global Control Register 0	Go
8h	CRC_CTRL1	CRC Global Control Register 1	Go
10h	CRC_CTRL2	CRC Global Control Register 2	Go
18h	CRC_INTS	CRC Interrupt Enable Set Register	Go
20h	CRC_INTR	CRC Interrupt Enable Reset Register	Go
28h	CRC_STATUS_REG	CRC Interrupt Status Register-	Go
30h	CRC_INT_OFFSET_REG	CRC Interrupt Offset	Go
38h	CRC_BUSY	CRC Busy Register during AUTO mode	Go
40h	CRC_PCOUNT_REG1	CRC Pattern Counter Pre-load Register1	Go
44h	CRC_SCOUNT_REG1	CRC Sector Counter Pre-load Register1	Go
48h	CRC_CURSEC_REG1	CRC Current Sector Regis-ter 1	Go
4Ch	CRC_WDTOPLD1	CRC channel 1 Watchdog Timeout Preload Register A	Go
50h	CRC_BCTOPLD1	CRC channel 1 Block Complete Timeout Preload Register B	Go
60h	PSA_SIGREGL1	Channel 1 PSA signature low register	Go
64h	PSA_SIGREGH1	Channel 1 PSA signature high register	Go
68h	CRC_REGL1	Channel 1 CRC value low register	Go
6Ch	CRC_REGH1	Channel 1 CRC value high register	Go
70h	PSA_SECSIGREGL1	Channel 1 PSA sector sig-nature low register	Go
74h	PSA_SECSIGREGH1	Channel 1 PSA sector sig-nature high register	Go
78h	RAW_DATAREGL1	Channel 1 Raw Data Low Register	Go
7Ch	RAW_DATAREGH1	Channel 1 Raw Data High Register	Go
80h	CRC_PCOUNT_REG2	CRC Pattern Counter Pre-load Register2	Go
84h	CRC_SCOUNT_REG2	CRC Sector Counter Pre-load Register2	Go
88h	CRC_CURSEC_REG2	CRC Current Sector Regis-ter 2	Go
8Ch	CRC_WDTOPLD2	CRC channel 2 Watchdog Timeout Preload Register	Go
90h	CRC_BCTOPLD2	CRC channel 2 Block Com-plete Timeout Preload Reg-ister	Go
A0h	PSA_SIGREGL2	Channel 2 PSA signature low register	Go
A4h	PSA_SIGREGH2	Channel 2 PSA signature high register	Go
A8h	CRC_REGL2	Channel 2 CRC value low register	Go
ACh	CRC_REGH2	Channel 2 CRC value high register	Go
B0h	PSA_SECSIGREGL2	Channel 2 PSA sector sig-nature low register	Go
B4h	PSA_SECSIGREGH2	Channel 2 PSA sector sig-nature high register	Go
B8h	RAW_DATAREGL2	Channel 2 Raw Data Low Register	Go
BCh	RAW_DATAREGH2	Channel 2 Raw Data High register	Go
C0h	CRC_PCOUNT_REG3	CRC Pattern Counter Pre-load Register3	Go
C4h	CRC_SCOUNT_REG3	CRC Sector Counter Pre-load Register3	Go
C8h	CRC_CURSEC_REG3	CRC Current Sector Regis-ter 3	Go
CCh	CRC_WDTOPLD3	CRC channel 3 Watchdog Timeout Preload Register	Go
D0h	CRC_BCTOPLD3	CRC channel 3 Block Com-plete Timeout Preload Reg-ister	Go
E0h	PSA_SIGREGL3	Channel 3 PSA signature low register	Go

Table 22-131. APP_CRC Registers (continued)

Offset	Acronym	Register Name	Section
E4h	PSA_SIGREGH3	Channel 3 PSA signature high register	Go
E8h	CRC_REGL3	Channel 3 CRC value low register	Go
ECh	CRC_REGH3	Channel 3 CRC value high register	Go
F0h	PSA_SECSIGREGL3	Channel 3 PSA sector sig-nature low register	Go
F4h	PSA_SECSIGREGH3	Channel 3 PSA sector sig-nature high register	Go
F8h	RAW_DATAREGL3	Channel 3 Raw Data Low Register	Go
FCh	RAW_DATAREGH3	Channel 3 Raw Data High register	Go
100h	CRC_PCOUNT_REG4	CRC Pattern Counter Pre-load Register4	Go
104h	CRC_SCOUNT_REG4	CRC Sector Counter Pre-load Register4	Go
108h	CRC_CURSEC_REG4	CRC Current Sector Regis-ter 4	Go
10Ch	CRC_WDTOPLD4	CRC channel 4 Watchdog Timeout Preload Register	Go
110h	CRC_BCTOPLD4	CRC channel 4 Block Com-plete Timeout Preload Reg-ister	Go
120h	PSA_SIGREGL4	Channel 4 PSA signature low register	Go
124h	PSA_SIGREGH4	Channel 4 PSA signature high register	Go
128h	CRC_REGL4	Channel 4 CRC value low register	Go
12Ch	CRC_REGH4	Channel 4 CRC value high register	Go
130h	PSA_SECSIGREGL4	Channel 4 PSA sector sig-nature low register	Go
134h	PSA_SECSIGREGH4	Channel 4 PSA sector sig-nature high register	Go
138h	RAW_DATAREGL4	Channel 4 Raw Data Low Register	Go
13Ch	RAW_DATAREGH4	Channel 4 Raw Data High register	Go
140h	MCRC_BUS_SEL	Data bus tracing selection	Go
144h	MCRC_RESERVED	RESERVED	Go

Complex bit access types are encoded to fit into small table cells. Table 22-132 shows the codes that are used for access types in this section.

Table 22-132. APP_CRC 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

22.5.6.1 CRC_CTRL0 Register (Offset = 0h) [Reset = 00000000h]

CRC_CTRL0 is shown in Table 22-133.

Return to the Summary Table.

Contains sw reset control bit to reset PSA

Table 22-133. CRC_CTRL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NU12	R	0h	Reserved
30	NU11	R	0h	Reserved
29	NU10	R	0h	Reserved

Table 22-133. CRC_CTRL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28-27	NU9	R	0h	Reserved
26-25	NU8	R	0h	Reserved
24	NU7	R	0h	Reserved
23	NU6	R	0h	Reserved
22	NU5	R	0h	Reserved
21	NU4	R	0h	Reserved
20-19	NU3	R	0h	Reserved
18-17	NU2	R	0h	Reserved
16	NU1	R	0h	Reserved
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 – Not Supported 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 – Not Supported 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

22.5.6.2 CRC_CTRL1 Register (Offset = 8h) [Reset = 00000000h]

CRC_CTRL1 is shown in Table 22-134.

Return to the Summary Table.

Contains power down control bit

Table 22-134. 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

22.5.6.3 CRC_CTRL2 Register (Offset = 10h) [Reset = 00000000h]

CRC_CTRL2 is shown in Table 22-135.

Return to the Summary Table.

Contains channel mode, data trace enable control bits

Table 22-135. CRC_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-24	NU14	R	0h	Reserved
23-18	RESERVED	R	0h	
17-16	NU13	R	0h	Reserved
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	

Table 22-135. CRC_CTRL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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

22.5.6.4 CRC_INTS Register (Offset = 18h) [Reset = 00000000h]

CRC_INTS is shown in Table 22-136.

Return to the Summary Table.

Write one to a bit to enable a interrupt

Table 22-136. CRC_INTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	NU22	R	0h	Reserved
27	NU21	R	0h	Reserved
26	NU20	R	0h	Reserved
25	NU19	R	0h	Reserved
24-21	RESERVED	R	0h	
20	NU18	R	0h	Reserved
19	NU17	R	0h	Reserved
18	NU16	R	0h	Reserved
17	NU15	R	0h	Reserved
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

Table 22-136. CRC_INTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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
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	

22.5.6.5 CRC_INTR Register (Offset = 20h) [Reset = 00000000h]

CRC_INTR is shown in Table 22-137.

Return to the Summary Table.

Write one to a bit to disable a interrupt

Table 22-137. CRC_INTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	NU30	R	0h	Reserved
27	NU29	R	0h	Reserved

Table 22-137. CRC_INTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	NU28	R	0h	Reserved
25	NU27	R	0h	Reserved
24-21	RESERVED	R	0h	
20	NU26	R	0h	Reserved
19	NU25	R	0h	Reserved
18	NU24	R	0h	Reserved
17	NU23	R	0h	Reserved
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

Table 22-137. CRC_INTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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
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	

22.5.6.6 CRC_STATUS_REG Register (Offset = 28h) [Reset = 00000000h]

CRC_STATUS_REG is shown in Table 22-138.

Return to the Summary Table.

Contains interrupt flags for different types of interrupt

Table 22-138. CRC_STATUS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	NU38	R	0h	Reserved
27	NU37	R	0h	Reserved
26	NU36	R	0h	Reserved
25	NU35	R	0h	Reserved
24-21	RESERVED	R	0h	
20	NU34	R	0h	Reserved
19	NU33	R	0h	Reserved
18	NU32	R	0h	Reserved
17	NU31	R	0h	Reserved
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 22-138. 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	

22.5.6.7 CRC_INT_OFFSET_REG Register (Offset = 30h) [Reset = 00000000h]

CRC_INT_OFFSET_REG is shown in Table 22-139.

Return to the Summary Table.

Contains the interrupt offset vector address

Table 22-139. CRC_INT_OFFSET_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	

Table 22-139. CRC_INT_OFFSET_REG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	OFSTREG	R/W	0h	CRC Interrupt Offset. This register indicates the highest priority pending interrupt vector address. Reading the offset register auto- matically clear the respective interrupt flag. Please reference Table 1–3. for details.

22.5.6.8 CRC_BUSY Register (Offset = 38h) [Reset = 00000000h]

CRC_BUSY is shown in Table 22-140.

Return to the Summary Table.

Contains the busy flag for each channel

Table 22-140. CRC_BUSY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	NU40	R	0h	Reserved
23-17	RESERVED	R	0h	
16	NU39	R	0h	Reserved
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.

22.5.6.9 CRC_PCOUNT_REG1 Register (Offset = 40h) [Reset = 00000000h]

CRC_PCOUNT_REG1 is shown in Table 22-141.

Return to the Summary Table.

Channel 1 preload register for the pattern count

Table 22-141. 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.

22.5.6.10 CRC_SCOUNT_REG1 Register (Offset = 44h) [Reset = 00000000h]

CRC_SCOUNT_REG1 is shown in Table 22-142.

Return to the Summary Table.

Channel 1 preload register for the sector count

Table 22-142. 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.

22.5.6.11 CRC_CURSEC_REG1 Register (Offset = 48h) [Reset = 00000000h]

CRC_CURSEC_REG1 is shown in Table 22-143.

Return to the Summary Table.

Channel 1 current sector register contains the sector number which causes CRC failure

Table 22-143. 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.

22.5.6.12 CRC_WDTOPLD1 Register (Offset = 4Ch) [Reset = 00000000h]

CRC_WDTOPLD1 is shown in Table 22-144.

Return to the Summary Table.

Channel 1 timeout pre-load value to check if within a given time DMA initiates a block transfer

Table 22-144. 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.

22.5.6.13 CRC_BCTOPLD1 Register (Offset = 50h) [Reset = 00000000h]

CRC_BCTOPLD1 is shown in Table 22-145.

Return to the Summary Table.

Channel 1 timeout pre-load value to check if one block of patterns are compressed with a given time

Table 22-145. CRC_BCTOPLD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	

Table 22-145. CRC_BCTOPLD1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

22.5.6.14 PSA_SIGREGL1 Register (Offset = 60h) [Reset = 00000000h]

PSA_SIGREGL1 is shown in Table 22-146.

Return to the Summary Table.

Channel 1 PSA signature low register

Table 22-146. 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.

22.5.6.15 PSA_SIGREGH1 Register (Offset = 64h) [Reset = 00000000h]

PSA_SIGREGH1 is shown in Table 22-147.

Return to the Summary Table.

Channel 1 PSA signature high register

Table 22-147. 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.

22.5.6.16 CRC_REGL1 Register (Offset = 68h) [Reset = 00000000h]

CRC_REGL1 is shown in Table 22-148.

Return to the Summary Table.

Channel 1 CRC value low register

Table 22-148. 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.

22.5.6.17 CRC_REGH1 Register (Offset = 6Ch) [Reset = 00000000h]

CRC_REGH1 is shown in Table 22-149.

Return to the Summary Table.

Channel 1 CRC value high register

Table 22-149. 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] regis- ter.

22.5.6.18 PSA_SECSIGREGL1 Register (Offset = 70h) [Reset = 00000000h]

PSA_SECSIGREGL1 is shown in Table 22-150.

Return to the Summary Table.

Channel 1 PSA sector signature low regis-ter

Table 22-150. 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.

22.5.6.19 PSA_SECSIGREGH1 Register (Offset = 74h) [Reset = 00000000h]

PSA_SECSIGREGH1 is shown in Table 22-151.

Return to the Summary Table.

Channel 1 PSA sector signature high regis-ter

Table 22-151. 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.

22.5.6.20 RAW_DATAREGL1 Register (Offset = 78h) [Reset = 00000000h]

RAW_DATAREGL1 is shown in Table 22-152.

Return to the Summary Table.

Channel 1 un-compressed raw data low register

Table 22-152. 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.

22.5.6.21 RAW_DATAREGH1 Register (Offset = 7Ch) [Reset = 00000000h]

RAW_DATAREGH1 is shown in Table 22-153.

Return to the Summary Table.

Channel 1 un-compressed raw data high register

Table 22-153. 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.

22.5.6.22 CRC_PCOUNT_REG2 Register (Offset = 80h) [Reset = 00000000h]

CRC_PCOUNT_REG2 is shown in Table 22-154.

Return to the Summary Table.

Channel 2 preload register for the pattern count

Table 22-154. 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.

22.5.6.23 CRC_SCOUNT_REG2 Register (Offset = 84h) [Reset = 00000000h]

CRC_SCOUNT_REG2 is shown in Table 22-155.

Return to the Summary Table.

Channel 2 preload register for the sector count

Table 22-155. 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.

22.5.6.24 CRC_CURSEC_REG2 Register (Offset = 88h) [Reset = 00000000h]

CRC_CURSEC_REG2 is shown in Table 22-156.

Return to the Summary Table.

Channel 2 current sector register contains the sector number which causes CRC fail-ure

Table 22-156. 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.

22.5.6.25 CRC_WDTOPLD2 Register (Offset = 8Ch) [Reset = 00000000h]

CRC_WDTOPLD2 is shown in Table 22-157.

Return to the Summary Table.

Channel 2 timeout pre-load value to check if within a given time DMA initiates a block transfer

Table 22-157. CRC_WDTOPLD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	

Table 22-157. CRC_WDTOPLD2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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.

22.5.6.26 CRC_BCTOPLD2 Register (Offset = 90h) [Reset = 00000000h]

CRC_BCTOPLD2 is shown in Table 22-158.

Return to the Summary Table.

Channel 2 timeout pre-load value to check if one block of patterns are compressed with a given time

Table 22-158. 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.

22.5.6.27 PSA_SIGREGL2 Register (Offset = A0h) [Reset = 00000000h]

PSA_SIGREGL2 is shown in Table 22-159.

Return to the Summary Table.

Channel 2 PSA signature low register

Table 22-159. 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.

22.5.6.28 PSA_SIGREGH2 Register (Offset = A4h) [Reset = 00000000h]

PSA_SIGREGH2 is shown in Table 22-160.

Return to the Summary Table.

Channel 2 PSA signature high register

Table 22-160. 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.

22.5.6.29 CRC_REGL2 Register (Offset = A8h) [Reset = 00000000h]

CRC_REGL2 is shown in Table 22-161.

Return to the Summary Table.

Channel 2 CRC value low register

Table 22-161. 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.

22.5.6.30 CRC_REGH2 Register (Offset = ACh) [Reset = 00000000h]

CRC_REGH2 is shown in Table 22-162.

Return to the Summary Table.

Channel 2 CRC value high register

Table 22-162. 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] regis- ter.

22.5.6.31 PSA_SECSIGREGL2 Register (Offset = B0h) [Reset = 00000000h]

PSA_SECSIGREGL2 is shown in Table 22-163.

Return to the Summary Table.

Channel 2 PSA sector signature low regis-ter

Table 22-163. 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.

22.5.6.32 PSA_SECSIGREGH2 Register (Offset = B4h) [Reset = 00000000h]

PSA_SECSIGREGH2 is shown in Table 22-164.

Return to the Summary Table.

Channel 2 PSA sector signature high regis-ter

Table 22-164. 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.

22.5.6.33 RAW_DATAREGL2 Register (Offset = B8h) [Reset = 00000000h]

RAW_DATAREGL2 is shown in Table 22-165.

Return to the Summary Table.

Channel 2 un-compressed raw data low register

Table 22-165. 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.

22.5.6.34 RAW_DATAREGH2 Register (Offset = BCh) [Reset = 00000000h]

RAW_DATAREGH2 is shown in Table 22-166.

Return to the Summary Table.

Channel 2 un-compressed raw data high Register

Table 22-166. 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.

22.5.6.35 CRC_PCOUNT_REG3 Register (Offset = C0h) [Reset = 00000000h]

CRC_PCOUNT_REG3 is shown in Table 22-167.

Return to the Summary Table.

Channel 3 preload register for the pattern count

Table 22-167. CRC_PCOUNT_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	NU41	R	0h	Reserved

22.5.6.36 CRC_SCOUNT_REG3 Register (Offset = C4h) [Reset = 00000000h]

CRC_SCOUNT_REG3 is shown in Table 22-168.

Return to the Summary Table.

Channel 3 preload register for the sector count

Table 22-168. CRC_SCOUNT_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU42	R	0h	Reserved

22.5.6.37 CRC_CURSEC_REG3 Register (Offset = C8h) [Reset = 00000000h]

CRC_CURSEC_REG3 is shown in Table 22-169.

Return to the Summary Table.

Channel 3 current sector register contains the sector number which causes CRC fail-ure

Table 22-169. CRC_CURSEC_REG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU43	R	0h	Reserved

22.5.6.38 CRC_WDTOPLD3 Register (Offset = CCh) [Reset = 00000000h]

CRC_WDTOPLD3 is shown in Table 22-170.

Return to the Summary Table.

Channel 3 timeout pre-load value to check if within a given time DMA initiates a block transfer

Table 22-170. CRC_WDTOPLD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU44	R	0h	Reserved

22.5.6.39 CRC_BCTOPLD3 Register (Offset = D0h) [Reset = 00000000h]

CRC_BCTOPLD3 is shown in Table 22-171.

Return to the Summary Table.

Channel 3 timeout pre-load value to check if one block of patterns are compressed with a given time

Table 22-171. CRC_BCTOPLD3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU45	R	0h	Reserved

22.5.6.40 PSA_SIGREGL3 Register (Offset = E0h) [Reset = 00000000h]

PSA_SIGREGL3 is shown in Table 22-172.

Return to the Summary Table.

Channel 3 PSA signature low register

Table 22-172. PSA_SIGREGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU46	R	0h	Reserved

22.5.6.41 PSA_SIGREGH3 Register (Offset = E4h) [Reset = 00000000h]

PSA_SIGREGH3 is shown in Table 22-173.

Return to the Summary Table.

Channel 3 PSA signature high register

Table 22-173. PSA_SIGREGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU47	R	0h	Reserved

22.5.6.42 CRC_REGL3 Register (Offset = E8h) [Reset = 00000000h]

CRC_REGL3 is shown in Table 22-174.

Return to the Summary Table.

Channel 3 CRC value low register

Table 22-174. CRC_REGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU48	R	0h	Reserved

22.5.6.43 CRC_REGH3 Register (Offset = ECh) [Reset = 00000000h]

CRC_REGH3 is shown in Table 22-175.

Return to the Summary Table.

Channel 3 CRC value high register

Table 22-175. CRC_REGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU49	R	0h	Reserved

22.5.6.44 PSA_SECSIGREGL3 Register (Offset = F0h) [Reset = 00000000h]

PSA_SECSIGREGL3 is shown in Table 22-176.

Return to the Summary Table.

Channel 3 PSA sector signature low regis-ter

Table 22-176. PSA_SECSIGREGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU50	R	0h	Reserved

22.5.6.45 PSA_SECSIGREGH3 Register (Offset = F4h) [Reset = 00000000h]

PSA_SECSIGREGH3 is shown in Table 22-177.

Return to the Summary Table.

Channel 3 PSA sector signature high regis-ter

Table 22-177. PSA_SECSIGREGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU51	R	0h	Reserved

22.5.6.46 RAW_DATAREGL3 Register (Offset = F8h) [Reset = 00000000h]

RAW_DATAREGL3 is shown in Table 22-178.

Return to the Summary Table.

Channel 3 un-compressed raw data low register

Table 22-178. RAW_DATAREGL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU52	R	0h	Reserved

22.5.6.47 RAW_DATAREGH3 Register (Offset = FCh) [Reset = 00000000h]

RAW_DATAREGH3 is shown in Table 22-179.

Return to the Summary Table.

Channel 3 un-compressed raw data high Register

Table 22-179. RAW_DATAREGH3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU53	R	0h	Reserved

22.5.6.48 CRC_PCOUNT_REG4 Register (Offset = 100h) [Reset = 00000000h]

CRC_PCOUNT_REG4 is shown in Table 22-180.

Return to the Summary Table.

Channel 4 preload register for the pattern count

Table 22-180. CRC_PCOUNT_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	NU54	R	0h	Reserved

22.5.6.49 CRC_SCOUNT_REG4 Register (Offset = 104h) [Reset = 00000000h]

CRC_SCOUNT_REG4 is shown in Table 22-181.

Return to the Summary Table.

Channel 4 preload register for the sector count

Table 22-181. CRC_SCOUNT_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU55	R	0h	Reserved

22.5.6.50 CRC_CURSEC_REG4 Register (Offset = 108h) [Reset = 00000000h]

CRC_CURSEC_REG4 is shown in Table 22-182.

Return to the Summary Table.

Channel 4 current sector register contains the sector number which causes CRC fail-ure

Table 22-182. CRC_CURSEC_REG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU56	R	0h	Reserved

22.5.6.51 CRC_WDTOPLD4 Register (Offset = 10Ch) [Reset = 00000000h]

CRC_WDTOPLD4 is shown in Table 22-183.

Return to the Summary Table.

Channel 4 timeout pre-load value to check if within a given time DMA initiates a block transfer

Table 22-183. CRC_WDTOPLD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU57	R	0h	Reserved

22.5.6.52 CRC_BCTOPLD4 Register (Offset = 110h) [Reset = 00000000h]

CRC_BCTOPLD4 is shown in Table 22-184.

Return to the Summary Table.

Channel 4 timeout pre-load value to check if one block of patterns are compressed with a given time

Table 22-184. CRC_BCTOPLD4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU58	R	0h	Reserved

22.5.6.53 PSA_SIGREGL4 Register (Offset = 120h) [Reset = 00000000h]

PSA_SIGREGL4 is shown in Table 22-185.

Return to the Summary Table.

Channel 4 PSA signature low register

Table 22-185. PSA_SIGREGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU59	R	0h	Reserved

22.5.6.54 PSA_SIGREGH4 Register (Offset = 124h) [Reset = 00000000h]

PSA_SIGREGH4 is shown in Table 22-186.

Return to the Summary Table.

Channel 4 PSA signature high register

Table 22-186. PSA_SIGREGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU60	R	0h	Reserved

22.5.6.55 CRC_REGL4 Register (Offset = 128h) [Reset = 00000000h]

CRC_REGL4 is shown in Table 22-187.

Return to the Summary Table.

Channel 4 CRC value low register

Table 22-187. CRC_REGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU61	R	0h	Reserved

22.5.6.56 CRC_REGH4 Register (Offset = 12Ch) [Reset = 00000000h]

CRC_REGH4 is shown in Table 22-188.

Return to the Summary Table.

Channel 4 CRC value high register

Table 22-188. CRC_REGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU62	R	0h	Reserved

22.5.6.57 PSA_SECSIGREGL4 Register (Offset = 130h) [Reset = 00000000h]

PSA_SECSIGREGL4 is shown in Table 22-189.

Return to the Summary Table.

Channel 4 PSA sector signature low regis-ter

Table 22-189. PSA_SECSIGREGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU63	R	0h	Reserved

22.5.6.58 PSA_SECSIGREGH4 Register (Offset = 134h) [Reset = 00000000h]

PSA_SECSIGREGH4 is shown in Table 22-190.

Return to the Summary Table.

Channel 4 PSA sector signature high regis-ter

Table 22-190. PSA_SECSIGREGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU64	R	0h	Reserved

22.5.6.59 RAW_DATAREGL4 Register (Offset = 138h) [Reset = 00000000h]

RAW_DATAREGL4 is shown in Table 22-191.

Return to the Summary Table.

Channel 4 un-compressed raw data low register

Table 22-191. RAW_DATAREGL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU65	R	0h	Reserved

22.5.6.60 RAW_DATAREGH4 Register (Offset = 13Ch) [Reset = 00000000h]

RAW_DATAREGH4 is shown in Table 22-192.

Return to the Summary Table.

Channel 4 un-compressed raw data high Register

Table 22-192. RAW_DATAREGH4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU66	R	0h	Reserved

22.5.6.61 MCRC_BUS_SEL Register (Offset = 140h) [Reset = 00000007h]

MCRC_BUS_SEL is shown in Table 22-193.

Return to the Summary Table.

Disables either or all tracing of data buses

Table 22-193. MCRC_BUS_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	NU67	R	0h	Reserved
2	MEn	R/W	1h	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

Table 22-193. MCRC_BUS_SEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DTCMEn	R/W	1h	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	1h	ITCMEn. Enable/disables the tracing of instruction TCM 0: Tracing of ITCM bus has been disabled 1: Tracing of ITCM bus has been enabled

22.5.6.62 MCRC_RESERVED Register (Offset = 144h) [Reset = 00000000h]

MCRC_RESERVED is shown in Table 22-194.

Return to the Summary Table.

0x144 to 0x1FF is reserved area.

Table 22-194. MCRC_RESERVED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NU68	R	0h	0x144 to 0x1FF is reserved area.

22.6 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.

22.6.1 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.

Name	Frame Address (Hex) Start	Frame Address (Hex) End	Size	Description
TOP_PBIST	0x5C02_0000	0x5C02_01CC	204B	PBIST module configuration registers. This IP covers memory test for all memories excluding DSP L1P, L1D and associated TAG memories

22.6.1.1 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 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.

22.6.1.2 PBIST Block Diagram

Figure 22-36 illustrates the basic PBIST blocks and its wrapper logic for the device.

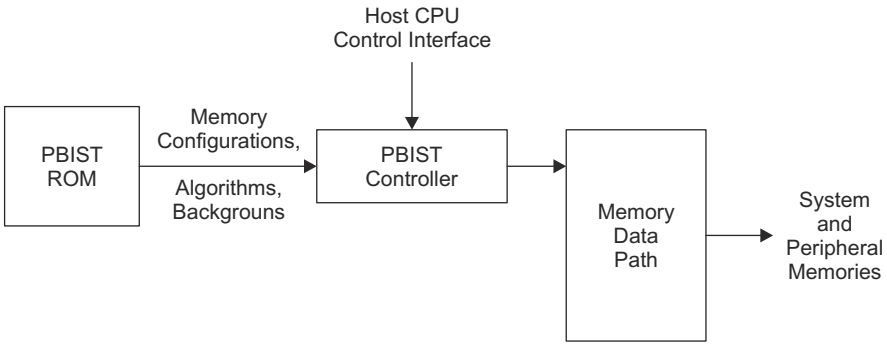


Figure 22-36. PBIST Block Diagram

22.6.1.2.1 On-chip ROM

The on-chip ROM contains the information regarding the algorithms and memories to be tested.

22.6.1.2.2 Host Processor Interface to the PBIST Controller Registers

The CPU can select the algorithm and RAM groups for the memories' self-test from the onchip 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.

22.6.1.2.3 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

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.

22.6.2 RAM/ROM Grouping and Algorithm

22.6.2.1 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 bits around the bit cell do not affect the bit cell.
- The bit cell can be written and read as both a 1 and a 0.

– 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

22.6.2.2 Read/write logic faults

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 worstcase 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

22.6.2.3

Note

March13N is the most recommended algorithm for the memory self-test

22.6.3 PBIST Registers

Table 22-195 lists the memory-mapped registers for the PBIST registers. All register offset addresses not listed in Table 22-195 should be considered as reserved locations and the register contents should not be modified.

Table 22-195. PBIST Registers

Offset	Acronym	Register Name	Section
0h	RESERVED	Reserved	Go
164h	PBIST_DLR	Datalogger 0	Go
16Ch	PBIST_PC	Program Control	Go
180h	PBIST_PACT	Pbist Active	Go
188h	PBIST_OVR	PBIST Overrides	Go
190h	PBIST_FSFR0	Fail status fail - port 0	Go
194h	PBIST_FSFR1	Fail status fail - port 1	Go
1C0h	PBIST_ROM	Rom Mask	Go
1C4h	PBIST_ALGO	ROM Algorithm Mask 0	Go
1C8h	PBIST_RINFOL	RAM Info Mask Lower 0	Go
1CCh	PBIST_RINFOU	RAM Info Mask Upper 0	Go

Complex bit access types are encoded to fit into small table cells. Table 22-196 shows the codes that are used for access types in this section.

Table 22-196. 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

22.6.3.1 RESERVED Register (Offset = 0h) [Reset = 00000000h]

RESERVED is shown in Table 22-197.

Return to the Summary Table.

Reserved

Table 22-197. RESERVED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

22.6.3.2 PBIST_DLR Register (Offset = 164h) [Reset = 0208h]

PBIST_DLR is shown in Table 22-198.

Return to the Summary Table.

Datalogger 0

Table 22-198. PBIST_DLR Register Field Descriptions

Bit	Field	Type	Reset	Description
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

22.6.3.3 PBIST_PC Register (Offset = 16Ch) [Reset = 00h]

PBIST_PC is shown in Table 22-199.

Return to the Summary Table.

Program Control

Table 22-199. PBIST_PC Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	PBIST_PC	R/W	0h	TI Internal Register.Reserved for HW RnD

22.6.3.4 PBIST_PACT Register (Offset = 180h) [Reset = 0h]

PBIST_PACT is shown in Table 22-200.

Return to the Summary Table.

Pbist Active

Table 22-200. PBIST_PACT Register Field Descriptions

Bit	Field	Type	Reset	Description
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

22.6.3.5 PBIST_OVR Register (Offset = 188h) [Reset = 0h]

PBIST_OVR is shown in Table 22-201.

Return to the Summary Table.

PBIST Overrides

Table 22-201. PBIST_OVR Register Field Descriptions

Bit	Field	Type	Reset	Description
3-0	PBIST_OVR	R/W	0h	TI Internal Register.Reserved for HW RnD

22.6.3.6 PBIST_FSFR0 Register (Offset = 190h) [Reset = 00000000h]

PBIST_FSFR0 is shown in Table 22-202.

Return to the Summary Table.

Fail status fail - port 0

Table 22-202. PBIST_FSFR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	TI Internal Register.Reserved for HW RnD
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

22.6.3.7 PBIST_FSFR1 Register (Offset = 194h) [Reset = 00000000h]

PBIST_FSFR1 is shown in Table 22-203.

Return to the Summary Table.

Fail status fail - port 1

Table 22-203. PBIST_FSFR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	TI Internal Register.Reserved for HW RnD
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

22.6.3.8 PBIST_ROM Register (Offset = 1C0h) [Reset = 3h]

PBIST_ROM is shown in Table 22-204.

Return to the Summary Table.

Rom Mask

Table 22-204. PBIST_ROM Register Field Descriptions

Bit	Field	Type	Reset	Description
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.

22.6.3.9 PBIST_ALGO Register (Offset = 1C4h) [Reset = FFFFFFFFh]

PBIST_ALGO is shown in Table 22-205.

Return to the Summary Table.

ROM Algorithm Mask 0

Table 22-205. PBIST_ALGO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	ALGO3	R/W	FFh	This register is used to indicate the algorithm(s) to be 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 algorithm(s) to be 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 algorithm(s) to be 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 algorithm(s) to be 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.

22.6.3.10 PBIST_RINFOL Register (Offset = 1C8h) [Reset = FFFFFFFFh]

PBIST_RINFOL is shown in Table 22-206.

Return to the Summary Table.

RAM Info Mask Lower 0

Table 22-206. 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 algorithmto 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 algorithmto 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 algorithmto 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.

Table 22-206. PBIST_RINFOL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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 algorithmto 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.

22.6.3.11 PBIST_RINFOU Register (Offset = 1CCh) [Reset = FFFFFFFFh]

PBIST_RINFOU is shown in Table 22-207.

Return to the Summary Table.

RAM Info Mask Upper 0

Table 22-207. 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 algorithmto 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 algorithmto 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 algorithmto 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 algorithmto 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 LPR Power Management

The xWRLx432 device supports stringent power requirements for consumer, broad-market applications. Power management techniques mentioned below are incorporated in the device and can help end-user achieve the defined goals.

23.1 Power Domains

The device is partitioned into mainly five power domains (as shown in Device Block Diagram), namely

•**RF/Analog Sub-System Power Domain**, which includes all the RF and Analog components required to transmit and receive the RF chirp signals

•**Front-End Controller Sub-System (FECSS) Power Domain,** which contains the ARM Cortex M3, responsible for radar front-end configuration, control, and calibration routines.

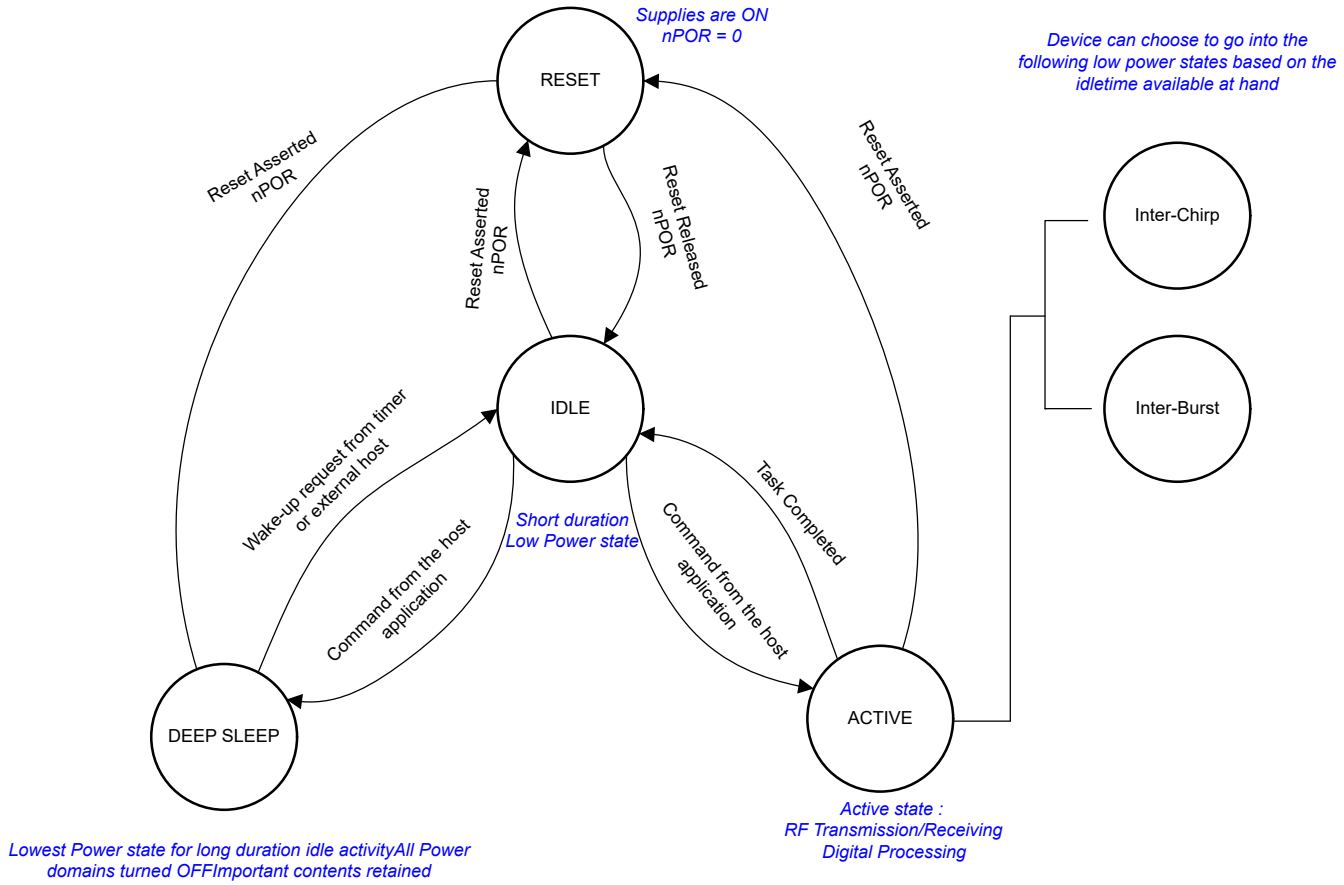
•**Application Sub-System (APPSS) Power Domain,** which includes a user programmable ARM Cortex M4, allowing for custom control and automotive interface applications. The Hardware Accelerator (HWA) block supplements the APPSS by offloading common radar processes, such as FFT, Constant False Alarm rate (CFAR), scaling, and compression. This sub-system also has controls for all the peripheral interfaces.

•**Hardware Accelerator Sub-System (HWASS) Power Domain,** which contains the radar hardware accelerator and the radar datacube memory, is responsible for pre-processing computations.

•And lastly, the **Test debug Sub-System(TESTDBG) Power Domain**, which contains the test and the debug logic that can be switched off in the field. This power domain is only required to be powered ON for test and debug functionality.

23.1.1 Power States

Based on the functional activity performed in the mmWave sensor, the device has the following power states:



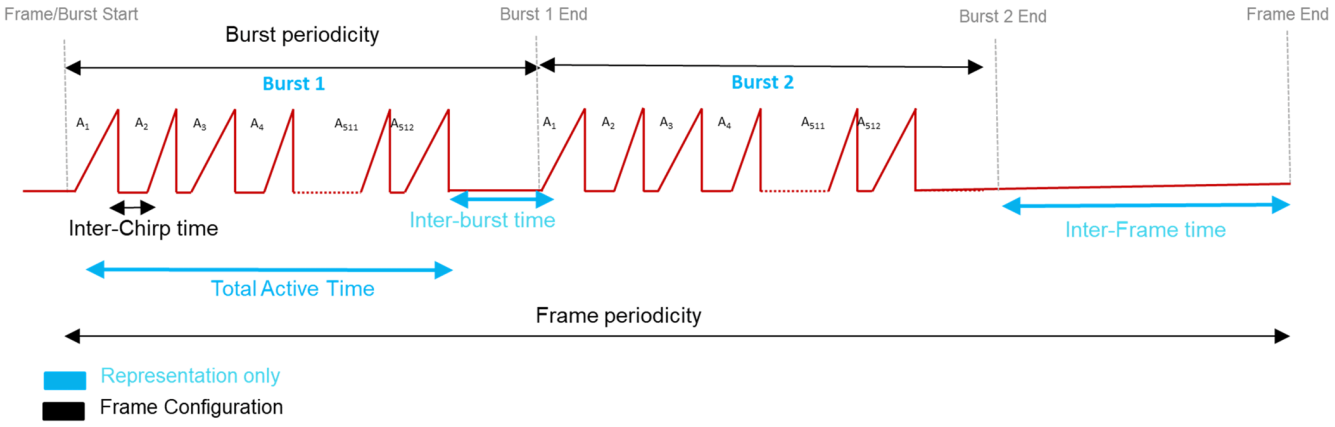
Active

The xWRLx432 mmWave Radar sensor detects objects and motion by emitting Frequency Modulated Continuous Wave (FMCW) chirps in the 57-64 GHz band of operation. The “Active” state is defined as when the device is chirping. In this state, the device can either be in a ‘Data Acquisition’ state, which is when data is being collected through the transmission and receipt of chirps, or in a ‘Data Processing’ state, when the samples recorded in the Data Acquisition state are being processed together.

During the Active state, most of the power domains are on, and the individual sections of the device are running at the highest clock frequency defined. The Active state is the highest power level state of the device. In operation, the device cycles in and out of active mode, emitting chirps with programmed idle durations in between.

Idle

As the name suggests, the Idle state occurs when the device is not actively chirping (transmitting or receiving FMCW chirps). Based on the application requirements and the idle time available, the device can choose to go into the following low power states.



1. **Inter-chirp Idle:** *Idle time in the order of a few us*
   a. This is the time between two consecutive chirps in the same burst. During this window, some of the Analog RF circuitry, including the TX power Amplifier, are turned off to save power.
   b. Power savings in the inter-chirp idle state are handled by the device firmware.
2. **Inter-burst Idle:** *Idle time in the order of a hundred us to a couple of ms*
   a. This is the time between two consecutive bursts in a frame. In this state, the device has slightly more idle time as compared to the inter-chirp idle time. Therefore, it can afford to save power by turning off a few more blocks than inter-chirp idle state does. More of the Analog RF circuitry, including the synthesizers, receiver and ADCs can be turned off to save a little more power burnout.
   b. Power savings in the inter-burst idle state are also handled by the device firmware.
3. **Inter-Frame Idle**: This is the remaining time in a frame after the device has completed the transmission and(or) processing of all chirp data. This is a completely application-driven state, capable of achieving far more savings that inter-chirp or inter-burst idle states.
   a. Based on the application requirements, the device can choose to go into a lower idle state or even into a sleep/deep sleep state as discussed further based on the time available at hand.
   b. In this mode, the device can also be waiting for a command from the external host or transferring the captured samples over a peripheral interface like SPI or CAN interface.

Sleep

This mode is for when the device does not have a task that needs to be immediately run, but will wake up in less than 10 milliseconds. This mode reduces power from the Interburst idle mode by turning off the Hardware Accelerator and clock-gating both the M4F Application Core and the M3F digital frontend core.

Deep Sleep

This is the lowest possible power state designed state in the device, where all the device power domains, including the Application sub-system (APPSS), along with Hardware Accelerator (HWA) and Front-End Controller system (FECSS) are powered off to save a significant amount of power.

Even though the entire device being is almost completely powered down, it does not need to reboot after waking up from deep sleep. The contents of the device, for instance, the Application Image, Chirp Profile etc. are retained across deep sleep cycles in the APPSS/FECSS memories.

Deep sleep exit is provisioned in the device through a number of external wakeup sources like UART/SPI/ GPIO/RTC/Sleep counter, etc.

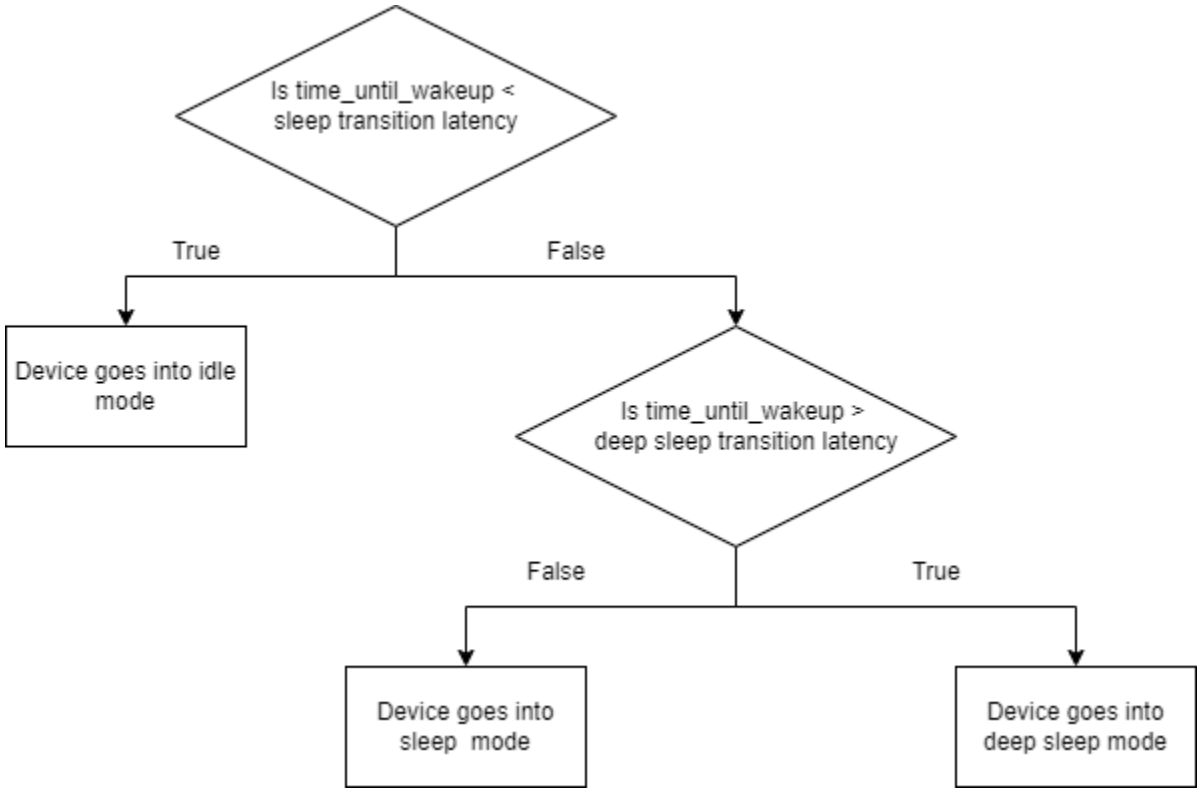


Figure 23-1.

More information about Power State transitions can be found in Section : Power State Transitions.

23.1.2 Power Sequence

Table 23-1. DEEP SLEEP Entry

Sr. No.	Step	Registers/Descriptions	Value
1	Release the DEEP SLEEP/ SLEEP entry pause register	Write the register only at the boot time of the device, not for every DEEP SLEEP entry. If this register has 0x0 value, then DEEP SLEEP entry is disabled. TOP_PRCM:RELEASE_PAUSE: RELEASE_PAUSE_RELEASE_PAUSE	0x1
2	Power down the RCOSC_10MHz	Do this step only once after power up to reduce the leakage current of RCOSC_10MHz	

Table 23-1. DEEP SLEEP Entry (continued)

Sr. No.	Step	Registers/Descriptions	Value
3	For debug mode only Remove the core reset for CM4/CM3 in DEEP SLEEP state	In DEEP SLEEP, the core reset of the CM4 and CM3 will be held low from PRCM. But in debug case after asserting forceactive, the core will still remain in reset because of the core resets. TOP_PRCM:APP_CORE_SYSRESET_PARAM TOP_PRCM:FEC_CORE_SYSRESET_PARAM	0x0001FFFF
4	Change IO cfg for DEEP SLEEP state	Add override value on IO's configured as outputs and driven by power down logic. This makes sure the IO's drive correct functional value and not the constant iso value. There is also possibility after DEEP SLEEP exit domains output (which is driving an IO output) can be 'x' which can be propagated to the output. TOP_IO_MUX:*	
5	Change all GCMs for subsystems and IPs to XTAL clock	Entering DEEP SLEEP state gates the XTAL clock and powers down the XTAL oscillator.	
6	Disable the PLLDIG clock and Power down APLL	If the PLLDIG is enabled, disable it before entering the DEEP SLEEP state as the XTAL oscillator will be powered down in DEEP SLEEP PLLDIG_CTRL:PLLDIG_EN:PLLDIG_EN_CFG_PLLDIG_EN	0x0
7	Configure the memory clusters retention	Set the memories SLEEP_STATE (for memory retention in DEEP SLEEP) and the ACTIVE_STATE (for memory state when device is in active state)	
8	Clear the SYS_RST_CAUSE register	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE_CLR	0x1
9	Wait till SYS_RST_CAUSE is clear	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE	0x0 (READ)
10	Enable SYS_RST_CAUSE register	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE_CLR	0x0
11	Power on and enable the RCOSC_10MHz, if needs to used for debugss clock in DEEP SLEEP	Only do this once if the RCOSC_10MHz is disabled.	

ADVANCE INFORMATION

Table 23-1. DEEP SLEEP Entry (continued)

Sr. No.	Step	Registers/Descriptions	Value
12	Change the debugss GCM clock source	During DEEP SLEEP, the XTAL clock is power down, so to provide core clock to debugss, change GCM to either RCOSC_10MHz or SLOW_CLK i.e. 32KHz clock TOP_PRCM:DEBUGSS_CLK_CLKCTL:DEBUGSS_CLK_CLKCTL_DEBUGSS_CLK_SRC_SEL	0x111 : RCOSC10M or 0x222 : SLOW_CLK
13	Set the wakeup source from DEEP SLEEP	TOP_PRCM:WU_SOURCE_EN:WU_SOURCE_EN_WU_SOURCE_EN	Set bit to 1, to enable the wakeup source for device DEEP SLEEP exit Bit 0 -> Sleep counter Bit 1 -> UART RX Bit 2 -> SPI CS Bit 3 -> GPIO or SYNCIN IO depends on the :WAKEUP_IO_MUX_SEL:WAKEUP_IO_MUX_SEL_WAKEUP_IO_MUX_SEL register Bit 4 -> RTC counter
14	Select the device deep sleep mode	APP_RCM:POWERMODE:POWERMODE_DEEPSLEEP	0x1
15	Set the M4 to enter DEEPSLEEP	Set bit SLEEPDEEP bit (bit 2) in System control register (0xE000ED10)	0x1
16	WFI	Execute the WFI instruction in M4 to take the device to DEEP SLEEP	

Table 23-2. Sleep Entry

Sr. No.	Step	Registers/Descriptions	Value
1	Release the DEEP SLEEP/ SLEEP entry pause register	Write this register only at the boot time of the device. Do not write for every SLEEP entry. If this register has 0x0 value then SLEEP entry is disabled. TOP_PRCM:RELEASE_PAUSE:RELEASE_PAUSE_RELEASE_PAUSE	0x1
2	Change all GCMs for subsystems and IPs to XTAL clock	The XTAL clock will be gated SLEEP state.	
3	Disable the PLLDIG clock	If the PLLDIG is enabled, disable it before entering the SLEEP state as the XTAL oscillator will be gated in SLEEP PLLDIG_CTRL:PLLDIG_EN:PLLDIG_EN_CFG_PLLDIG_EN	0x0
4	Clear the SYS_RST_CAUSE register	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE_CLR	0x1
5	Wait till SYS_RST_CAUSE is clear	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE	0x0 (READ)

Table 23-2. Sleep Entry (continued)

Sr. No.	Step	Registers/Descriptions	Value
6	Enable SYS_RST_CAUSE register	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE_CLR	0x0
7	Turn on the RCOSC_10MHz, if needs to used for debugss clock in DEEP SLEEP	TOP_PRCM:EFUSE_10M_OSC_DISABLE:EFUSE_10M_OSC_DISABLE_SEL_OV_EFUSE_10MHZ_OSC_DISABLE TOP_PRCM:EFUSE_10M_OSC_DISABLE:EFUSE_10M_OSC_DISABLE_OV_EFUSE_10MHZ_OSC_DISABLE	0x7 0x0
8	Change the debugss GCM clock source	During SLEEP the XTAL clock is gated. To provide core clock to debugss, change GCM to either RCOSC_10MHz or SLOW_CLK i.e. 32KHz clock TOP_PRCM:DEBUGSS_CLK_CLKCTL:DEBUGSS_CLK_CLKCTL_DEBUGSS_CLK_SRC_SEL	0x111 : RCOSC10M or 0x222 : SLOW_CLK
9	Set the wakeup source from SLEEP	TOP_PRCM:WU_SOURCE_EN:WU_SOURCE_EN_WU_SOURCE_EN	Set bit to 1, to enable the wakeup source for device SLEEP exit Bit 0 -> Sleep counter Bit 1 -> UART RX Bit 2 -> SPI CS Bit 3 -> GPIO or SYNCIN IO depends on the TOP_PRCM Registers:WAKEUP_IO_MUX_SEL:WAKEUP_IO_MUX_SEL_WAKEUP_IO_MUX_SEL register Bit 4 -> RTC counter Bit 5 -> FRC frame start intr (Use only for device SLEEP wakeup)
10	Select the device sleep mode	APP_RCM:POWERMODE:POWERMODE_SLEEP	0x1
11	Set the M4 to enter SLEEP	Clear bit SLEEPDEEP bit (bit 2) in System control register (0xE000ED10)	0x0
12	WFI	Execute the WFI instruction in M4 to take the device to DEEP SLEEP	

Table 23-3. Deep Sleep Exit

Sr. No.	Step	Registers/Descriptions	Value
1	Check SYS_RST_CAUSE	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE	0x0 (READ)
2	Get the wakeup status	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS	0x2 (READ)

ADVANCE INFORMATION

Table 23-3. Deep Sleep Exit (continued)

Sr. No.	Step	Registers/Descriptions	Value
3	Get the wakeup source	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_SOURCE	If the corresponding bit is 1, it indicates the source cause the wakeupBit 0 -> Sleep counter as Wakeup sourceBit 1 -> UART as Wakeup sourceBit 2 -> SPI as Wakeup sourceBit 3 -> GPIO as Wakeup sourceBit 4 -> RTC counter as Wakeup source
4	Clear the wakeup status and source register	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS_CLEAR	0x1
5	Wait till wakeup status and source is clear	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS	0x0 (READ)
6	Enable wakeup status and source register	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS_CLEAR	0x0
7	Change the debugss GCM clock source to TOPSS CLOCK	TOP_PRCM:DEBUGSS_CLK_CLKCTL:DEBUGSS_CLK_CLKCTL_DEBUGSS_CLK_SRC_SEL	0x0 : TOPSS_SYS_CLK
8	Turn off the RCOSC_10MHz	TOP_PRCM:EFUSE_10M_OSC_DISABLE:EFUSE_10M_OSC_DISABLE_OV_EFUSE_10MHZ_OSC_DISABLE	0x1

Note

Do not power up all three domains (APPSS, FECSS, HWASS) at the same time during DEEP SLEEP exit. If all three domains are powered up at the same time, the inrush current on VDDAR net causes high IR drop on the memories which are retained. The high IR drop causes the loss of memory data in the retains memories.

Power the three domains in a staggered manner using one of the below methods.

1. Before entering DEEP SLEEP, keep the FECSS and HWASS power domains in manually power down state. After DEEP SLEEP exit, power them up one by one.
   a. This is the recommended way, as device can enter DEEP SLEEP only after idle state. In idle state the FECSS and HWASS domains stay powered off.
2. Using PWR_REQ_PARAM register:
   a. In the PWR_REQ_PARAM registers, program the field WAKEUP_DELAY_COUNT to stagger the wakeup of domain in DEEP SLEEP exit.
   b. WAKEUP_DELAY_COUNT value can be less than TOP_PRCM:WU_COUNTER_END

Table 23-4. Sleep Exit

Sr. No.	Step	Registers/Descriptions	Value
1	Check SYS_RST_CAUSE	TOP_PRCM:SYS_RST_CAUSE:SYS_RST_CAUSE_SYS_RST_CAUSE	0x0 (READ)
2	Get the wakeup status	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS	0x1 (READ)

Table 23-4. Sleep Exit (continued)

Sr. No.	Step	Registers/Descriptions	Value
3	Get the wakeup source	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_SOURCE	If the corresponding bit is 1, it indicates the source cause the wakeupBit 0 -> Sleep counter as Wakeup sourceBit 1 -> UART as Wakeup sourceBit 2 -> SPI as Wakeup sourceBit 3 -> GPIO as Wakeup sourceBit 4 -> RTC counter as Wakeup sourceBit 5 -> FRC frame start intr as Wakeup source
4	Clear the wakeup status and source register	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS_CLEAR	0x1
5	Wait until wakeup status and source is clear	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS	0x0 (READ)
6	Enable wakeup status and source register	TOP_PRCM:RADAR_WAKEUP_STATUS:RADAR_WAKEUP_STATUS_WAKEUP_STATUS_CLEAR	0x0
7	Change the debugss GCM clock source to TOPSS CLOCK	TOP_PRCM:DEBUGSS_CLK_CLKCTL:DEBUGSS_CLK_CLKCTL_DEBUGSS_CLK_SRC_SEL	0x0 : TOPSS_SYS_CLK
8	Turn off the RCOSC_10MHz	TOP_PRCM:EFUSE_10M_OSC_DISABLE:EFUSE_10M_OSC_DISABLE_OV_EFUSE_10MHZ_OSC_DISABLE	0x1

23.1.3 APPSS Power Domains

23.1.3.1 Overview

The APPSS power domain cannot be taken to power down state alone or forced power down. Only in DEEP SLEEP state it can enter the power down state.

23.1.3.2 CPU/Device Sleep/Deepsleep

Low Power Mode	CM4 Output ports		APPSS RCM REG		Exit Criteria	Entry	Description	CM3*	CM4+
GATE DEEPSLEEP MODE	SLEEPING	SLEEPDEEP	POWERMODE_SLEEP	POWERMODE_DEEPSLEEP					
CPU SLEEP MODE	1	0	1	1	Any interrupt on NVIC	Executing WFI(1) instruction and CPSR[SLEEPDEEP] is cleared.	HCLK to the CPU GATED. Power is saved internally to the core. PRCM is not involved.	YES	YES
CPU DEEPSLEEP MODE	1	1	1	1	WIC Signal - see Section 23.7	Executing WFI instruction along with setting CPSR[SLEEPDEEP] of CM4.	Both HCLK and FCLK to the CPU are gated. PRCM is not involved.	YES	YES

ADVANCE INFORMATION

Low Power Mode	CM4 Output ports		APPSS RCM REG		Exit Criteria	Entry	Description	CM3*	CM4+
DEVICE SLEEP MODE	1	0	0	1	Wakeup Events are defined in Section 23.6	Executing WFI instruction and CPSR[SLEEPDEEP] is cleared.	Handled by PRCM. Can be entered only by CM4.	NA	YES
DEVICE DEEPSLEEP MODE	1	1	1	0	Wakeup Events are defined in Section 23.6	Executing WFI instruction along with setting CPSR[SLEEPDEEP] of CM4.	Handled by PRCM. Can be entered only by CM4.	NA	YES

(1) WFE instruction is also used for entering sleep but the wakeup mechanism is not implemented using EVENTO/EVENTI.

23.1.3.3 M4 CPU Sleep/Deepsleep Sequence

Table 23-5. CPU DEEP SLEEP Entry

Sr. No.	Step	Registers/Descriptions	Value
1	Select the CPU deep sleep mode	APP_RCM:POWERMODE:POWERMODE_DEEPSLEEP	0x0
2	Set the M4 to enter DEEPSLEEP	Set bit SLEEPDEEP bit (bit 2) in System control register (0xE000ED10)	0x1
3	WFI	Execute the WFI instruction in M4 to take the device to DEEP SLEEP	

Table 23-6. CPU SLEEP Entry

Sr. No.	Step	Registers/Descriptions	Value
1	Select the CPU sleep mode	APP_RCM:POWERMODE:POWERMODE_SLEEP	0x0
2	Set the M4 to enter SLEEP	Clear bit SLEEPDEEP bit (bit 2) in System control register (0xE000ED10)	0x0
3	WFI	Execute the WFI instruction in M4 to take the device to DEEP SLEEP	

23.1.3.4 M3 CPU Sleep/Deepsleep Sequence

Table 23-7. CPU DEEP SLEEP Entry

Sr. No.	Step	Registers/Descriptions	Value
1	Set the M3 to enter DEEPSLEEP	Set bit SLEEPDEEP bit (bit 2) in System control register (0xE000ED10)	0x1
2	WFI	Execute the WFI instruction in M3 to take the device to DEEP SLEEP	

Table 23-8. CPU SLEEP Entry

Sr. No.	Step	Registers/Descriptions	Value
1	Set the M3 to enter SLEEP	Clear bit SLEEPDEEP bit (bit 2) in System control register (0xE000ED10)	0x0
2	WFI	Execute the WFI instruction in M3 to take the device to DEEP SLEEP	

23.1.4 FECSS Power Domain

23.1.4.1 Overview

The FECSS sub system can be taken powered down independently of the APPSS and HWASS. The CM4 can force power down or power up of the FECSS domain. This is handled by the DFP APIs

There are three memory clusters in the FECSS sub system. Each of the memory clusters can be configured to be powered up or powered down separately in both the domain's power up or power down state.

The FECSS sub system can configured in one of the following three states,

Table 23-9.

State	HWASS Power State	Memory Clusters Power State	Comment
1	ON	ALL clusters ON	Default configuration, Reset state of domain
2	ON	SOME or ALL clusters OFF	To save power unused memories are powered off
3	OFF	ALL clusters OFF	Default configuration
4	OFF	SOME or ALL clusters ON	Cluster kept for memory retention

23.1.4.2 Power Sequence

Table 23-10.

Initial State	Final State	SEQUENCE
FECSS ON ALL clusters ON **(1)**	FECSS OFF ALL clusters OFF & SOME or ALL clusters ON **(3) & (4)**	1. Write the memory SLEEP_STATE and ACTIVE_STATE registers depending on the memory state required in power down and in after power up (refer "Memory Clusters and Grouping" table for registers) 2. Change the FECSS domain power control to manual (This step can be done at done at the start once, no need to repeat every time) a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_MODE = 0x0 3. Power down the FECSS domain a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x0 4. Poll the FECSS power status register, if the 96Kb shared memory in the FECSS is used by HWASS then FECSS domain wont be power down in this case use the RESET status register a. HWASS Domains is also power down, i. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_POWER_STATUS (READ) 0x0 b. HWASS Domains is also power up, i. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_RESET_STATUS (READ) 0x0
FECSS OFF **(3) & (4)**	FECSS ON **(1) & (2)**	1. Power up the FECSS domain a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x1 2. Poll the FECSS power status register, a. HWASS Domains is also power down, i. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_POWER_STATUS (READ) 0x1 b. HWASS Domains is also power up, i. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_RESET_STATUS (READ) 0x1

Table 23-10. (continued)

Initial State	Final State	SEQUENCE
FECSS ON ALL clusters ON **(1)**	FECSS ON SOME or ALL clusters OFF **(2)**	1. Write the memory SLEEP_STATE and ACTIVE_STATE registers depending on the memory state required in power down and in after power up (refer "Memory Clusters and Grouping" table for registers) 2. Change the FECSS domain power control to manual (This step can be done at done at the start once, no need to repeat every time) a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_MODE = 0x0 3. Power down the FECSS domain a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x0 4. Poll the FECSS power status register, a. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_POWER_STATUS (READ) 0x0 5. Power up the FECSS domain a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x1 6. Poll the FECSS power status register, a. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_POWER_STATUS (READ) 0x1

Table 23-10. (continued)

Initial State	Final State	SEQUENCE
FECSS ON SOME or ALL clusters OFF **(2)**	FECSS ON ALL clusters ON & SOME or ALL clusters OFF **(1) & (2)**	1. Write the memory SLEEP_STATE and ACTIVE_STATE registers depending on the memory state required in power down and in after power up (refer "Memory Clusters and Grouping" table for registers) 2. Change the FECSS domain power control to manual (This step can be done at done at the start once, no need to repeat every time) a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_MODE = 0x0 3. Power down the FECSS domain a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x0 4. Poll the FECSS power status register, a. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_POWER_STATUS (READ) 0x0 5. Power up the FECSS domain a. TOP_PRCM:FEC_PWR_REQ_PARAM:FEC_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x1 6. Poll the FECSS power status register, a. TOP_PRCM:PSCON_FEC_PD_EN:PSCON_FEC_PD_EN_FEC_PD_POWER_STATUS (READ) 0x1

Careabouts for FECSS Power Domain:

Errors and interrupts coming from FECSS MPU are unmasked by default. Need to mask them to prevent x propagation when FECSS powers up again.

- APP_CTRL:APPSS_MPU_ERRAGG_MASK_FECSS_MPU

Writing 1'b1 masks the corresponding interrupt/error.

23.1.5 HWASS Power Domains

23.1.5.1 Overview

The HWASS sub system can be taken powered down independently of the APPSS and FECSS. The CM4 can force power down or power up of the HWASS domain.

There are three memory clusters in HWASS sub system. Each of the memory clusters can be configured to be power up or power down separately in both the HWASS domain's power up or down state.

The HWASS sub system can configured in one of the following three states:

Table 23-11.

State	HWASS Power State	Memory Clusters Power State	Comment
1	ON	ALL clusters ON	Default configuration, Reset state of domain
2	ON	SOME or ALL clusters OFF	To save power unused memories are powered off

Table 23-11. (continued)

State	HWASS Power State	Memory Clusters Power State	Comment
3	OFF	ALL clusters OFF	Default configuration
4	OFF	SOME or ALL clusters ON	Cluster kept for memory retention

23.1.5.2 Power Sequence

Table 23-12.

Initial State	Final State	SEQUENCE
HWASS ON ALL clusters ON **(1)**	HWASS OFF ALL clusters OFF & SOME or ALL clusters ON **(3) & (4)**	1. Write the memory SLEEP_STATE and ACTIVE_STATE registers depending on the memory state required in power down and in after power up 2. Change the HWASS domain power control to manual (This step can be done at done at the start once, no need to repeat every time) a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_MODE = 0x0 3. Power down the HWASS domain a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x0 4. Poll the HWASS power status register, a. TOP_PRCM:PSCON_HWA_PD_EN:PSCON_HWA_PD_EN_HWA_PD_POWER_STATUS (READ) 0x0
HWASS OFF **(3) & (4)**	HWASS ON **(1) & (2)**	1. Power up the HWASS domain a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x1 2. Poll the HWASS power status register, a. TOP_PRCM:PSCON_HWA_PD_EN:PSCON_HWA_PD_EN_HWA_PD_POWER_STATUS (READ) 0x1

Table 23-12. (continued)

Initial State	Final State	SEQUENCE
HWASS ON ALL clusters ON **(1)**	HWASS ON SOME or ALL clusters OFF **(2)**	1. Write the memory SLEEP_STATE and ACTIVE_STATE registers depending on the memory state required in power down and in after power up 2. Change the HWASS domain power control to manual (This step can be done at done at the start once, no need to repeat every time) a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_MODE = 0x0 3. Power down the HWASS domain a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x0 4. Poll the HWASS power status register, a. TOP_PRCM:PSCON_HWA_PD_EN:PSCON_HWA_PD_EN_HWA_PD_POWER_STATUS (READ) 0x0 5. Power up the HWASS domain a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x1 6. Poll the HWASS power status register,TOP_PRCM:PSCON_HWA_PD_EN:PSCON_HWA_PD_EN_HWA_PD_POWER_STATUS (READ) 0x1

Table 23-12. (continued)

Initial State	Final State	SEQUENCE
HWASS ON SOME or ALL clusters OFF **(2)**	HWASS ON ALL clusters ON & SOME or ALL clusters OFF **(1) & (2)**	1. Write the memory SLEEP_STATE and ACTIVE_STATE registers depending on the memory state required in power down and in after power up 2. Change the HWASS domain power control to manual (This step can be done at done at the start once, no need to repeat every time) a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_MODE = 0x0 3. Power down the HWASS domain a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x0 4. Poll the HWASS power status register, a. TOP_PRCM:PSCON_HWA_PD_EN:PSCON_HWA_PD_EN_HWA_PD_POWER_STATUS (READ) 0x0 5. Power up the HWASS domain a. TOP_PRCM:HWA_PWR_REQ_PARAM:HWA_PWR_REQ_PARAM_WAKEUP_OUT_STATE = 0x1 6. Poll the HWASS power status register, a. TOP_PRCM:PSCON_HWA_PD_EN:PSCON_HWA_PD_EN_HWA_PD_POWER_STATUS (READ) 0x1

Careabouts for HWASS Power Domain:

a.Poll the "HWASS_LOOP_INT" connected to APPSS Interrupts (CORTEX M4 Interrupt Map).

The HWASS_LOOP_INT goes high when all HWA operations are complete.

b. Errors and interrupts coming from tptc/tpcc in HWASS, HWA IP and shared memory are unmasked by default. Need to mask them to prevent propagation when HWASS powers up again.

The mask registers for each interrupts and errors are at :-

APP_CTRL:APPSS_TPCC_B_ERRAGG_MASK

APP_CTRL:APPSS_TPCC_B_INTAGG_MASK

HWA_CFG:HWA_SAFETY_ERR_MASK

APP_HWA_ADCBUF_CTRL:HWASS_SHRD_RAM_ACCESS_ERROR_MASK

Writing 1'b1 masks the corresponding interrupt/error.

APP_CTRL:HW_SPARE_RW0:HW_SPARE_RW0_HW_SPARE_RW0

Bit 0: Writing 1'b1 masks the hwa local ram agg serr propagation to ESM

Writing 1'b0 unmasks the hwa local ram agg serr propagation to ESM

Bit 1 : Writing 1'b1 masks the hwa local ram agg uerr propagation to ESM

Writing 1'b0 unmasks the hwa local ram agg uerr propagation to ESM

23.1.6 TESTDBG Power Domain

23.1.6.1 Overview

The TESTDBG power domain can be powered down or powered up irrespective of other power domain state. The power controls of the TESTDBG are not controlled through the RADAR POWER STATE FSM like the other power domains. i.e. the DEEP SLEEP entry/exit of the device does not affect the power state of TESTDBG domain.

The power state of TESTDBG is controlled through 2 mechanisms,

1. ICE-Melter : After connecting the debugger the ICE-Melter generates the power request for the TESTDBG power domain.
2. Override register

In the following three states TESTDBG domain can configured,

Table 23-13.

Sr. No.	State	Source Control
1	Force Power up	Override register
2	Force Power Down	Override Register
3	ICE-Melter control	ICE-Melter

23.1.6.2 Power Sequence

Table 23-14.

Initial State	Final State	SEQUENCE
TESTDBG ON Force Power up **(1)**	TESTDBG OFF Force Power Down **(2)**	1. Power down using overrides a. TOP_PRCM:PSCON_TEST_DBG_PD_EN:PSCON_TEST_DBG_PD_EN_OV_TEST_DBG_PD_IS_SLEEP = 0x1 2. Poll the TESTDBG power status register, a. TOP_PRCM:PSCON_TEST_DBG_PD_EN:PSCON_TEST_DBG_PD_EN_TEST_DBG_PD_POWER_STATUS (READ) 0x0
TESTDBG ON Force Power up **(1)**	TESTDBG ON/OFF ICE-Melter control **(3)**	1. Shift the controls to ICE-Melter a. TOP_PRCM:PSCON_TEST_DBG_PD_EN:PSCON_TEST_DBG_PD_EN_SEL_OV_TEST_DBG_PD_IS_SLEEP = 0x0 2. Poll the TESTDBG power status register, a. TOP_PRCM:PSCON_TEST_DBG_PD_EN:PSCON_TEST_DBG_PD_EN_TEST_DBG_PD_POWER_STATUS (READ)

ADVANCE INFORMATION

Table 23-14. (continued)

Initial State	Final State	SEQUENCE
TESTDBG OFF Force Power Down **(2)**	TESTDBG ON/OFF ICE-Melter control **(3)**	Shift the controls to ICE-Melter 1. TOP_PRCM:PSCON_TEST_DBG_PD_EN:PSCON_TEST_DBG_PD_EN_SEL_OV_TEST_DBG_PD_IS_SLEEP = 0x0 2. Poll the TESTDBG power status register, a. TOP_PRCM:PSCON_TEST_DBG_PD_EN:PSCON_TEST_DBG_PD_EN_TEST_DBG_PD_POWER_STATUS

23.1.7 ANALOG Power Domain

Analog transmit and receive sub-systems are turned off when not chirping. This is implemented in the mmwave DFP APIs.

23.1.8 Memory Power Domains

23.1.8.1 Memory Clusters and Grouping

	CLUSTER #	MEMORY CLUSTERS	RETAINABLE (R) or NOT-RETAINABLE (NR)	Memory power state control Registers*
APP_PD	**#1**	64KB BANK#1 (RAM_1A)	R (Not on VDDAR SWITCH)	TOP_PRCM:PSCON_APP_PD_RAM_STATE:PSCON_APP_PD_RAM_STATE_APP_PD_MEM_SLEEP_STATE[0] TOP_PRCM:PSCON_APP_PD_RAM_STATE:PSCON_APP_PD_RAM_STATE_APP_PD_MEM_ACTIVE_STATE[0]
	#2	16KB BANK#2 (RAM_2A), RAM 2KB	R (Not on VDDAR SWITCH)	TOP_PRCM:PSCON_APP_PD_RAM_STATE:PSCON_APP_PD_RAM_STATE_APP_PD_MEM_SLEEP_STATE[1] TOP_PRCM:PSCON_APP_PD_RAM_STATE:PSCON_APP_PD_RAM_STATE_APP_PD_MEM_ACTIVE_STATE[1]
	#3	64KB BANK#1 (RAM_1B)	R (Not on VDDAR SWITCH)	TOP_PRCM:PSCON_APP_PD_RAM_STATE:PSCON_APP_PD_RAM_STATE_APP_PD_MEM_SLEEP_STATE[2] TOP_PRCM:PSCON_APP_PD_RAM_STATE:PSCON_APP_PD_RAM_STATE_APP_PD_MEM_ACTIVE_STATE[2]
	#4	128KB BANK#1 (RAM_1C)	R	TOP_PRCM:PSCON_APP_PD_RAM_GRP1_STATE:PSCON_APP_PD_RAM_GRP1_STATE_APP_PD_MEM_GRP1_SLEEP_STATE[0] TOP_PRCM:PSCON_APP_PD_RAM_GRP1_STATE:PSCON_APP_PD_RAM_GRP1_STATE_APP_PD_MEM_GRP1_ACTIVE_STATE[0]
	#5	112KB BANK#2 (RAM_2B), 128KB BANK#3 (RAM_3)	R	TOP_PRCM:PSCON_APP_PD_RAM_GRP1_STATE:PSCON_APP_PD_RAM_GRP1_STATE_APP_PD_MEM_GRP1_SLEEP_STATE[1] TOP_PRCM:PSCON_APP_PD_RAM_GRP1_STATE:PSCON_APP_PD_RAM_GRP1_STATE_APP_PD_MEM_GRP1_ACTIVE_STATE[1]
	#6	256KB SHARED RAM (APP_SHMEM_1 , APP_SHMEM_2)	R	TOP_PRCM:PSCON_APP_PD_RAM_GRP2_STATE:PSCON_APP_PD_RAM_GRP2_STATE_APP_PD_MEM_GRP2_SLEEP_STATE TOP_PRCM:PSCON_APP_PD_RAM_GRP2_STATE:PSCON_APP_PD_RAM_GRP2_STATE_APP_PD_MEM_GRP2_ACTIVE_STATE
FECSS_PD	**#1**	FEC_RAM_1A 16KB BANK#1, RAMPGEN RAMs	R (Not on VDDAR SWITCH)	TOP_PRCM:PSCON_FEC_PD_RAM_STATE:PSCON_FEC_PD_RAM_STATE_FEC_PD_MEM_SLEEP_STATE TOP_PRCM:PSCON_FEC_PD_RAM_STATE:PSCON_FEC_PD_RAM_STATE_FEC_PD_MEM_ACTIVE_STATE
	#2	FEC_RAM_1B 16KB BANK#2, DFE, BISTFFT, GPADC	R	TOP_PRCM:PSCON_FEC_PD_RAM_GRP4_STATE:PSCON_FEC_PD_RAM_GRP4_STATE_FEC_PD_MEM_GRP4_SLEEP_STATE[0] TOP_PRCM:PSCON_FEC_PD_RAM_GRP4_STATE:PSCON_FEC_PD_RAM_GRP4_STATE_FEC_PD_MEM_GRP4_ACTIVE_STATE[0]
	#3	96KB SHARED RAM	R	TOP_PRCM:PSCON_FEC_PD_RAM_GRP4_STATE:PSCON_FEC_PD_RAM_GRP4_STATE_FEC_PD_MEM_GRP4_SLEEP_STATE[1] TOP_PRCM:PSCON_FEC_PD_RAM_GRP4_STATE:PSCON_FEC_PD_RAM_GRP4_STATE_FEC_PD_MEM_GRP4_ACTIVE_STATE[1]

	CLUSTER #	MEMORY CLUSTERS	RETAINABLE (R) or NOT-RETAINABLE (NR)	Memory power state control Registers*
HWA_PD	**#1**	HWA PARAM RAM	R	TOP_PRCM:PSCON_HWA_PD_RAM_GRP3_STATE:PSCON_HWA_PD_RAM_GRP3_STATE_HWA_PD_MEM_GRP3_SLEEP_STATE[0] TOP_PRCM:PSCON_HWA_PD_RAM_GRP3_STATE:PSCON_HWA_PD_RAM_GRP3_STATE_HWA_PD_MEM_GRP3_ACTIVE_STATE[0]
	#2	FFT ENGINE RAM, LOCAL BUFFERS	R	TOP_PRCM:PSCON_HWA_PD_RAM_GRP3_STATE:PSCON_HWA_PD_RAM_GRP3_STATE_HWA_PD_MEM_GRP3_SLEEP_STATE[1] TOP_PRCM:PSCON_HWA_PD_RAM_GRP3_STATE:PSCON_HWA_PD_RAM_GRP3_STATE_HWA_PD_MEM_GRP3_ACTIVE_STATE[1]
	#3	160KB SHARED RAM	R	TOP_PRCM:PSCON_HWA_PD_RAM_GRP3_STATE:PSCON_HWA_PD_RAM_GRP3_STATE_HWA_PD_MEM_GRP3_SLEEP_STATE[2] TOP_PRCM:PSCON_HWA_PD_RAM_GRP3_STATE:PSCON_HWA_PD_RAM_GRP3_STATE_HWA_PD_MEM_GRP3_ACTIVE_STATE[2]

*Writing to the ACTIVE_STATE and SLEEP_STATE registers for the memory clusters won't change the memory clusters' power state immediately. To change the power state of the memory clusters according to these register values, take the respective power domain through power cycle, i.e. power down and power up, of domain.

Note

1. GPADC data memory (output buffer) should not be kept in retention state in DEEP SLEEP or in fecss power down. There is a possibility this memory can get corrupted during the DEEP SLEEP exit or FECSS power up due to x propagation on the EZ control.
2. When Memory is power down SW should make sure it does not access the memory
3. When the internal LDO's SRAM and DIG supply is used, then do not remove the DFTRTA overrides controls, By default the DFTRTA controls are kept in low state using override (TOP_PRCM:PSCON_DFTRTA_OVERRIDE)
   a. Only when external 1.2V supply is present the DFTRTA overrides can be removed and let the DFTRTA signals to be driven from mem PSCON.
   b. In case of external supply present even if we keep the overrides on DFTRTA signals, the only drawback is we will have higher leakage.

23.1.8.2 Memory Power Control Override

	CLUSTER #	MEMORY CLUSTERS	RETAINABLE (R) or NOT-RETAINABLE (NR)	Memory power state control Registers*
APP_PD	**#1**	64KB BANK#1 (RAM_1A)	R (Not on VDDAR SWITCH)	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_AGOODIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_AGOODIN[0]
	#2	16KB BANK#2 (RAM_2A), RAM 2KB	R (Not on VDDAR SWITCH)	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_AGOODIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_AGOODIN[1]
	#3	64KB BANK#1 (RAM_1B)	R (Not on VDDAR SWITCH)	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_AONIN[2] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_AONIN[2] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_AGOODIN[2] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_AGOODIN[2]
	#4	128KB BANK#1 (RAM_1C)	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_GRP1_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_GRP1_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_GRP1_AGOODIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_GRP1_AGOODIN[0]
	#5	112KB BANK#2 (RAM_2B), 128KB BANK#3 (RAM_3)	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_GRP1_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_GRP1_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_GRP1_AGOODIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_GRP1_AGOODIN[1]
	#6	256KB SHARED RAM (APP_SHMEM_1 , APP_SHMEM_2)	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_GRP2_AONIN TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_GRP2_AONIN TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_1:DEBUG_MEM_PSCON_OVERRIDE_SEL_1_SEL_OV_APP_PD_MEM_GRP2_AGOODIN TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_1:DEBUG_MEM_PSCON_OVERRIDE_VAL_1_OV_APP_PD_MEM_GRP2_AGOODIN

	CLUSTER #	MEMORY CLUSTERS	RETAINABLE (R) or NOT-RETAINABLE (NR)	Memory power state control Registers*
FECSS_PD	**#1**	FEC_RAM_1A 16KB BANK#1, RAMPGEN RAMs	R (Not on VDDAR SWITCH)	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_FEC_PD_MEM_AONIN TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_FEC_PD_MEM_AONIN TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_FEC_PD_MEM_AGOODIN TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_FEC_PD_MEM_AGOODIN
	#2	FEC_RAM_1B 16KB BANK#2, DFE, BISTFFT, GPADC	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_FEC_PD_MEM_GRP4_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_FEC_PD_MEM_GRP4_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_FEC_PD_MEM_GRP4_AGOODIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_FEC_PD_MEM_GRP4_AGOODIN[0]
	#3	96KB SHARED RAM	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_FEC_PD_MEM_GRP4_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_FEC_PD_MEM_GRP4_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_FEC_PD_MEM_GRP4_AGOODIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_FEC_PD_MEM_GRP4_AGOODIN[1]
HWA_PD	**#1**	HWA PARAM RAM	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_HWA_PD_MEM_GRP3_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_HWA_PD_MEM_GRP3_AONIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_HWA_PD_MEM_GRP3_AGOODIN[0] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_HWA_PD_MEM_GRP3_AGOODIN[0]
	#2	FFT ENGINE RAM, LOCAL BUFFERS	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_HWA_PD_MEM_GRP3_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_HWA_PD_MEM_GRP3_AONIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_HWA_PD_MEM_GRP3_AGOODIN[1] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_HWA_PD_MEM_GRP3_AGOODIN[1]
	#3	160KB SHARED RAM	R	TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_HWA_PD_MEM_GRP3_AONIN[2] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_HWA_PD_MEM_GRP3_AONIN[2] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_SEL_2:DEBUG_MEM_PSCON_OVERRIDE_SEL_2_SEL_OV_HWA_PD_MEM_GRP3_AGOODIN[2] TOP_PRCM:DEBUG_MEM_PSCON_OVERRIDE_VAL_2:DEBUG_MEM_PSCON_OVERRIDE_VAL_2_OV_HWA_PD_MEM_GRP3_AGOODIN[2]

Sequencing Requirement for Memory power control overrides,

1. Memory Power down
   a. AONIN and AGOODIN can be made low simultaneously, no sequencing required
   b. The only condition is if AONIN is 1 and AGOODIN is 0, then there is no access to memory
2. Memory Power up
   a. First, turn on AONIN. After approximately 30us, turn on AGOODIN
   b. If AGOODIN comes up first or with AONIN, the device can be damaged due to high inrush current.

Note

This sequencing is properly handled by mem pscon. Use the ACTIVE STATE and SLEEP STATE register to change the power state of memories.

23.2 IO Power Management

This section describes the IO states (output/input, if output - what is the value, pulldown/pullup) during the following conditions

- SLEEP/DEEPSLEEP scenarios
- RESET state - For the System topologies where the Radar device is kept in reset while the external host is up.
- BOOT time (as it last for ~100ms) - To reduce the Energy consumption in periodic power-up usecases.

Device State	IO Pull	IO_Overrides (OE)
SLEEP/DEEP_SLEEP STATE	Software shall configure the pulls as per system topology	Software shall configure the OEs and RX_ENABLEs disabled wherever possible • While entering the Deep Sleep mode, to save leakage power, most of the IOs will be put in output HighZ state, RXACTIVE needs to be enabled only for the wakeup source pins (and the TDI, TCK, TMS and RS232RX pins for development/debug) and Clock input pins (RTC_CLK_IN).
RESET STATE	Pull value is specific to peripheral used in a system topology. End-user should avoid leakage due to opposite state of Pulls	• IOs are set in high Z • For external components like QSPI, PMIC, EEPROM, Control Chip selects are in known state (disabled) during reset state. • Reset value is such that – all the Outputs are put in HighZ state (except RS232TX and TDO) using the OE_OVERRIDE and OE_OVERRIDE_CTL registers reset values – All RXs are disabled using IE_OVERRIDE and IE_OVERRIDE_CTL registers reset values except the TMS, TDI, TCK and RS232RX pins.

BOOTUP (At RESET Release)		• To keep the leakage in IOs low during the boot time, all the outputs would follow RESET state configuration by default and all RXs need to be “Disabled” except the TMS, TDI, TCK and RS232RX. TMS, TDI, TCK, RS232RX are required to talk to the device even before the boot rom comes to picture. (Precautionary- TMS, RS232RX are kept in PU state during reset and bootup time) • Once BOOT ROM picks up, software would search for only “boot peripherals” and keep on setting the configurations of Pulls (including QSPI flash mode), enables and directionality accordingly. If not found, it shall restore the RESET state of the these IOs • FUNC_SEL reset value selects the ‘Boot Peripheral Search’ mode as muxing option by default. Couple of exceptions as specified in FUNC_SEL column • Once system boots Bootloader (relevant for Image download) and Application would configure the FUNC_SEL and Pulls appropriately.

The table below shows the pin-muxing of the device and the ball state of the pads during the states mentioned above.

Table 23-15. Pin Muxing Table

BGA BALL NUMBER[1]	BALL NAME[2]	SIGNAL NAME[3]	PINCNTL REGISTER [4]	PIN CNTL REGISTER ADDRESS [5] [11]	MODE [6]	TYPE [7]	PULL UP/ DOWN TYPE[8]	BALL STATE DURING RST[9]	BALL STATE AFTER RST [10]
H10	GPIO_2	GPIO_2	PADAL_CFG_REG	0x5A00 002C	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		LIN_RX			1	I			
		WARM_RESET_OUT			2	O			
		I2C_SDA			3	IO			
		SPIA_CS1_N			4	IO			
		WU_REQIN			5	I			
		RTC_CLK_IN			6	I			
		MDO_D0			7	O			
J10	GPIO_5	GPIO_5	PADAV_CFG_REG	0x5A00 0054	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		SYNC_IN			1	I			
		LIN_RX			2	I			
		EPWMB			3	O			
		EPWM_SYNC_IN			4	I			
		MDO_D3			5	O			

ADVANCE INFORMATION

Table 23-15. Pin Muxing Table (continued)

BGA BALL NUMBER[1]	BALL NAME[2]	SIGNAL NAME[3]	PINCNTL REGISTER [4]	PIN CNTL REGISTER ADDRESS [5] [11]	MODE [6]	TYPE [7]	PULL UP/ DOWN TYPE[8]	BALL STATE DURING RST[9]	BALL STATE AFTER RST [10]
M10	HOST_CLK_REQ	HOST_CLK_REQ	PADAX_CFG_REG	0x5A00 005C	0	O	PU/PD	OFF/OFF/OFF	OFF/SS/OFF
		GPIO_7			1	IO			
		MCU_CLKOUT			2	O			
		LIN_TX			3	O			
		WU_REQIN			4	I			
		SPIB_MISO			5	IO			
		I2C_SCL			6	IO			
		MDO_D3			8	O			
		MDO_FRM_CLK			9	O			
K11	NERROR_OUT	NERROR_OUT	PADAU_CFG_REG	0x5A00 0050	0	O	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		GPIO_4			1	IO			
		SYNC_IN			2	I			
		SPIB_CS0_N			3	IO			
		WU_REQIN			4	I			
		RTC_CLK_IN			5	I			
		MCU_CLKOUT			6	O			
		MDO_D3			7	O			
H11	PMIC_CLKOUT	PMIC_CLKOUT	PADAK_CFG_REG	0x5A00 0028	0	O	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		LIN_TX			1	O			
		SPIA_CS1_N			2	IO			
		MDO_FRM_CLK			3	O			
B11	QSPI[0]	QSPI[0]	PADAC_CFG_REG	0x5A00 0008	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		SPIB_MOSI			1	IO			
		MDO_D0			2	O			
B8	QSPI[1]	QSPI[1]	PADAD_CFG_REG	0x5A00 000C	0	I	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		SPIB_MISO			1	IO			
		RTC_CLK_IN			2	I			
		MDO_D3			3	O			
B10	QSPI[2]	QSPI[2]	PADAE_CFG_REG	0x5A00 0010	0	I	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		I2C_SCL			1	IO			
		WU_REQIN			2	I			
		MDO_D1			3	O			
B9	QSPI[3]	QSPI[3]	PADAF_CFG_REG	0x5A00 0014	0	I	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		I2C_SDA			1	IO			
		SYNC_IN			2	I			
		MDO_D2			3	O			
A11	QSPI_CLK	QSPI_CLK	PADAA_CFG_REG	0x5A00 0000	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		SPIB_CLK			1	IO			
		MDO_CLK			2	O			
A10	QSPI_CS	QSPI_CS	PADAB_CFG_REG	0x5A00 0004	0	O	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		SPIB_CS0_N			1	IO			
		MDO_FRM_CLK			2	O			

ADVANCE INFORMATION

Table 23-15. Pin Muxing Table (continued)

BGA BALL NUMBER[1]	BALL NAME[2]	SIGNAL NAME[3]	PINCNTL REGISTER [4]	PIN CNTL REGISTER ADDRESS [5] [11]	MODE [6]	TYPE [7]	PULL UP/ DOWN TYPE[8]	BALL STATE DURING RST[9]	BALL STATE AFTER RST [10]
F11	RS232_RX	RS232_RX	PADAP_CFG_REG	0x5A00 003C	0	I	PU/PD	OFF/OFF/UP	OFF/OFF/UP
		I2C_SDA			1	IO			
		UARTB_RX			2	I			
		LIN_RX			3	I			
		MDO_D2			4	O			
		SPIB_MISO			5	IO			
E10	RS232_TX	RS232_TX	PADAO_CFG_REG	0x5A00 0038	0	O	PU/PD	OFF/OFF/OFF	OFF/OFF/OFFI
		I2C_SCL			1	IO			
		UARTB_TX			2	O			
		LIN_TX			3	O			
		EPWM_SYNC_IN			4	I			
		MDO_D1			5	O			
		SPIB_CS1_N			6	IO			
D10	SPIA_CLK	SPIA_CLK	PADAG_CFG_REG	0x5A00 0018	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		EPWMB			1	O			
		I2C_SCL			2	IO			
		SPIB_CLK			3	IO			
		MDO_CLK			4	O			
D11	SPIA_CS0_N	SPIA_CS0_N	PADAH_CFG_REG	0x5A00 001C	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		EPWMA			1	O			
		I2C_SDA			2	IO			
		SPIB_CS0_N			3	IO			
		MDO_D3			4	O			
C11	SPIA_MISO	SPIA_MISO	PADAJ_CFG_REG	0x5A00 0024	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		GPIO_1			1	IO			
		EPWMA			2	O			
		SPIB_MISO			3	IO			
		MDO_D2			4	O			
B12	SPIA_MOSI	SPIA_MOSI	PADAI_CFG_REG	0x5A00 0020	0	IO	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		GPIO_0			1	IO			
		EPWMB			2	O			
		SPIB_MOSI			3	IO			
		MDO_D1			4	O			
C12	TCK	TCK	PADAT_CFG_REG	0x5A00 004C	0	I	PU/PD	OFF/OFF/ DOWN	OFF/OFF/ DOWN
		EPWMB			1	O			
		SPIB_CS1_N			2	IO			
		SPIB_MOSI			3	IO			
		MDO_D0			4	O			
G11	TDI	TDI	PADAR_CFG_REG	0x5A00 0044	0	I	PU/PD	OFF/OFF/ DOWN	OFF/OFF/ DOWN
		EPWMA			1	O			
		SPIB_CS0_N			2	IO			
E11	TDO	TDO	PADAS_CFG_REG	0x5A00 0048	0	O	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		MDO_FRM_CLK			1	O			

ADVANCE INFORMATION

Table 23-15. Pin Muxing Table (continued)

BGA BALL NUMBER[1]	BALL NAME[2]	SIGNAL NAME[3]	PINCNTL REGISTER [4]	PIN CNTL REGISTER ADDRESS [5] [11]	MODE [6]	TYPE [7]	PULL UP/ DOWN TYPE[8]	BALL STATE DURING RST[9]	BALL STATE AFTER RST [10]
E12	TMS	TMS	PADAQ_CFG_REG	0x5A00 0040	0	I	PU/PD	OFF/OFF/UP	OFF/OFF/UP
		WARM_RESET_OUT			1	O			
		SPIA_CS1_N			2	IO			
		SYNC_IN			3	I			
		SPIB_MISO			4	IO			
		SPIB_CLK			5	IO			
		RTC_CLK_IN			6	I			
		EPWM_SYNC_IN			7	I			
		EPWM_SYNC_OUT			8	O			
L11	UARTA_RTS	UART_RTS	PADAW_CFG_REG	0x5A00 0058	0	O	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		GPIO_6			1	IO			
		LIN_TX			2	O			
		SPIB_CLK			3	IO			
		WU_REQIN			4	I			
		EPWMA			5	O			
		RTC_CLK_IN			6	I			
		MDO_CLK			7	O			
J11	UARTA_RX	UARTA_RX	PADAM_CFG_REG	0x5A00 0030	0	I	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		GPIO_3			1	IO			
		LIN_RX			2	I			
		CAN_FD_RX			3	I			
		SYNC_IN			4	I			
		UARTB_RX			5	I			
		I2C_SDA			6	IO			
		MDO_D1			7	O			
L12	UARTA_TX	UARTA_TX	PADAN_CFG_REG	0x5A00 0034	0	O	PU/PD	OFF/OFF/OFF	OFF/OFF/OFF
		LIN_TX			1	O			
		CAN_FC_TX			2	O			
		SPIB_MOSI			3	IO			
		WU_REQIN			4	I			
		UARTB_TX			5	O			
		I2C_SCL			6	IO			
		MDO_D2			7	O			

(1) **BALL NUMBER:** Ball numbers on the bottom side associated with each signal on the bottom.
(2) **BALL NAME:** Mechanical name from package device (name is taken from muxmode 0).
(3) **SIGNAL NAME:** Names of signals multiplexed on each ball (also notice that the name of the ball is the signal name in muxmode 0).
(4) **PINCNTL_REGISTER:** APPSS Register name for PinMux Control
(5) **PINCNTL ADDRESS:** APPSS Address for PinMux Control
(6) **MODE:** Multiplexing mode number: value written to PinMux Cntl register to select specific Signal name for this Ball number. Mode column has bit range value.
(7) **TYPE:** Signal type and direction:
- I = Input
- O = Output
- IO = Input or Output

(8) **PULL UP/DOWN TYPE:** indicates the presence of an internal pullup or pulldown resistor. Pullup and pulldown resistors can be enabled or disabled via software.
- Pull Up: Internal pullup

ADVANCE INFORMATION

- Pull Down: Internal pulldown
- An empty box means No pull.

(9) **BALL STATE DURING RST**: State of Ball during reset in the format of RX/TX/Pull Status
(10) **BALL STATE AFTER RST**: State of Ball after reset in the format of RX/TX/Pull Status
(11) Pin Mux Control Value maps to lower 4 bits of register.

23.3 Initial Power States

	COMPONENTS	RESET	BOOT
	PORG	OFF	ON
	AON PD (PRCM)	OFF	ON
ANA	SLICER LDO (XO+SLICER)	OFF	ON
	APLL VCO LDO, IOBUF LDO	OFF	OFF
	MDLL	OFF	OFF
	SYNTH VCO LDO, DIV_LDO, SDM_LDO	OFF	OFF
	RX FE, IFA, IFA SAT Detection, ADCs, DC_OFFSET_CORR	OFF	OFF
	IFA BIAS, ADC BIAS, RX BIAS, IFA LDOs, ADC DIG LDOs	OFF	OFF
	RF LDO (PA, LNA) , BIAS	OFF	OFF
FECSS	FEC SS PD State	ON	ON
			Default FEC PD will be ON state;To force on state write FEC_PWR_REQ_PARAM = 18'h217FF
	Cortex M3	OFF	PLL_DIG/4
			FEC_SYS_CLKCTL_SRCSEL = 0x333FEC_SYS_CLKCTL_DIVR = 0x444
	FECSS I/C, Peripherals	OFF	PLL_DIG/4
			FEC_SYS_CLKCTL_SRCSEL = 0x333FEC_SYS_CLKCTL_DIVR = 0x444
	DFE	OFF	CLK GATED
			FEC_RX_ADC_CLKCTL_GATE = 0x7IPCFGCLKGATE0_DFE_CFG = 0x7
	RAMPGEN	OFF	CLK GATED
			FEC_RX_ADC_CLKCTL_GATE = 0x7IPCFGCLKGATE0_RAMPGEN = 0x7
HWASS	HWASS PD State	ON	ON
			Default HWA PD will be ON state;To force on state write HWA_PWR_REQ_PARAM = 18'h217FF
	HWASS I/C, TPTCs, TPCCs	OFF	CLK GATED
	HWA 1.2	OFF	CLK GATED
TOP SS	FRC	OFF	CLK GATED
	PLL_DIG	OFF	ON
			PLLDIG_EN_CFG_PLLDIG_EN = 0x7

ADVANCE INFORMATION

	COMPONENTS	RESET	BOOT
APPSS	APPSS PD State	ON	ON
			Default APP PD will be ON state;To force on state write APP_PWR_REQ_PARAM = 18'h217FF
	Cortex M4F	OFF	PLL_DIG/2
			APP_CPU_CLKCTL_SRCSEL = 0x333APP_CPU_CLKCTL_DIVR = 0x222
	APPSS I/C, Peripherals	OFF	CPU_CLK/2
			APP_CPU_CLKCTL_SRCSEL = 0x333APP_CPU_CLKCTL_DIVR = 0x222
	CAN FD	OFF	CLK GATED
			APP_CAN_CLKCTL_GATE = 0x7

ADVANCE INFORMATION

23.4 Data Acquisition States

The four data acquisition states are - Data Acquisition, Inter-chirp Idle, Inter-burst Idle and Data Processing. They are represented as columns below.

	COMPONENTS	DATA ACQUISITION	INTERCHIRP IDLE	INTERBURST IDLE	DATA PROCESSING (Non-inline Interframe processing)
	PORG	ON	ON	ON	ON
	AON PD (PRCM)	ON	ON	ON	ON
ANA	SLICER LDO (XO+SLICER)	ON	ON	ON	ON
	APLL VCO LDO, IOBUF LDO	ON	ON	ON	OFF
	MDLL	OFF	OFF	OFF	OFF
	SYNTH VCO LDO, DIV_LDO, SDM_LDO	ON	ON	OFF	OFF
	RX FE, IFA, IFA SAT Detection, ADCs, DC_OFFSET_CORR	ON	ON* (<10us)	OFF	OFF
	IFA BIAS, ADC BIAS, RX BIAS, IFA LDOs, ADC DIG LDOs	ON	ON	OFF	OFF
	RF LDO (PA, LNA) , BIAS	ON	OFF	OFF	OFF

	COMPONENTS	DATA ACQUISITION	INTERCHIRP IDLE	INTERBURST IDLE	DATA PROCESSING (Non-inline Interframe processing)
FECSS	FEC SS PD State	ON	ON	ON	RETENTION
		Default FEC PD will be ON state;To force on state write FEC_PWR_REQ_PARAM = 18'h217FF	Default FEC PD will be ON state;To force on state write FEC_PWR_REQ_PARAM = 18'h217FF	Default FEC PD will be ON state;To force on state write FEC_PWR_REQ_PARAM = 18'h217FF	To force off state write FEC_PWR_REQ_PARAM = 18'h207FF
	Cortex M3	WFI, on XTAL CLK	WFI, on XTAL CLK	WFI, on XTAL CLK	OFF
		FEC_SYS_CLKCTL_SRCSEL = 0x000FEC_SYS_CLKCTL_DIVR = 0x000	FEC_SYS_CLKCTL_SRCSEL = 0x000FEC_SYS_CLKCTL_DIVR = 0x000	FEC_SYS_CLKCTL_SRCSEL = 0x000FEC_SYS_CLKCTL_DIVR = 0x000	Power Off
	FECSS I/C, Peripherals	On XTAL CLK	On XTAL CLK	On XTAL CLK	OFF
		FEC_SYS_CLKCTL_SRCSEL = 0x000FEC_SYS_CLKCTL_DIVR = 0x000	FEC_SYS_CLKCTL_SRCSEL = 0x000FEC_SYS_CLKCTL_DIVR = 0x000	FEC_SYS_CLKCTL_SRCSEL = 0x000FEC_SYS_CLKCTL_DIVR = 0x000	Power Off
	DFE	ON	DYNAMIC CLK GATED	DYNAMIC CLK GATED	OFF
		FEC_RX_ADC_CLKCTL_GATE = 0x0IPCFGCLKGATE0_DFE_CFG = 0x0	Not implemented	Not implemented	Power Off
	RAMPGEN	ON	ON	ON	OFF
		FEC_RX_ADC_CLKCTL_GATE = 0x0IPCFGCLKGATE0_RAMPGEN = 0x0	FEC_RX_ADC_CLKCTL_GATE = 0x0IPCFGCLKGATE0_RAMPGEN = 0x0	FEC_RX_ADC_CLKCTL_GATE = 0x0IPCFGCLKGATE0_RAMPGEN = 0x0	Power Off
HWASS	HWASS PD State	ON	ON	ON	ON
		Default HWA PD will be ON state;To force on state write HWA_PWR_REQ_PARAM = 18'h217FF	Default HWA PD will be ON state;To force on state write HWA_PWR_REQ_PARAM = 18'h217FF	Default HWA PD will be ON state;To force on state write HWA_PWR_REQ_PARAM = 18'h217FF	To force off state write HWA_PWR_REQ_PARAM = 18'h207FF
	HWASS I/C, TPTCs, TPCCs	on XTAL CLK/2	on XTAL CLK/2	on XTAL CLK/2	on PLL_DIG/4
		APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x333APP_CPU_CLKCTL_DIVR = 0x222
	HWA 1.2	CLK GATED	CLK GATED	CLK GATED	on PLL_DIG/4
					APP_CPU_CLKCTL_SRCSEL = 0x333APP_CPU_CLKCTL_DIVR = 0x222
TOP SS	FRC	XTAL CLK	XTAL CLK	XTAL CLK	XTAL CLK
	PLL_DIG	OFF	OFF	OFF	ON
		PLLDIG_EN_CFG_PLLDIG_EN = 0x0	PLLDIG_EN_CFG_PLLDIG_EN = 0x0	PLLDIG_EN_CFG_PLLDIG_EN = 0x0	PLLDIG_EN_CFG_PLLDIG_EN = 0x7

ADVANCE INFORMATION

	COMPONENTS	DATA ACQUISITION	INTERCHIRP IDLE	INTERBURST IDLE	DATA PROCESSING (Non-inline Interframe processing)
APPSS	APPSS PD State	ON	ON	ON	ON
		Default APP PD will be ON state;To force on state write APP_PWR_REQ_PARAM = 18'h217FF	Default APP PD will be ON state;To force on state write APP_PWR_REQ_PARAM = 18'h217FF	Default APP PD will be ON state;To force on state write APP_PWR_REQ_PARAM = 18'h217FF	To force off state write APP_PWR_REQ_PARAM = 18'h207FF
	Cortex M4F	WFI, on XTAL CLK	WFI, on XTAL CLK	WFI, on XTAL CLK	On PLL_DIG/2
		APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x333APP_CPU_CLKCTL_DIVR = 0x222
	APPSS I/C, Peripherals	on XTAL CLK/2	on XTAL CLK/2	on XTAL CLK/2	on PLL_DIG/4
		APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	APP_CPU_CLKCTL_SRCSEL = 0x333APP_CPU_CLKCTL_DIVR = 0x222
	CAN FD	OSC_CLKx2	OSC_CLKx2	OSC_CLKx2	OSC_CLKx2
		APP_CAN_CLKCTL_SRCSEL = 0x111APP_CAN_CLKCTL_DIVR = 0x000	APP_CAN_CLKCTL_SRCSEL = 0x111APP_CAN_CLKCTL_DIVR = 0x000	APP_CAN_CLKCTL_SRCSEL = 0x111APP_CAN_CLKCTL_DIVR = 0x000	APP_CAN_CLKCTL_SRCSEL = 0x111APP_CAN_CLKCTL_DIVR = 0x000

23.5 Power State Transitions.

The low power states are - Idle, Device Sleep and Device Deep Sleep. They are represented as columns below.

	COMPONENTS	IDLE	DEVICE SLEEP (<10ms)	DEVICE DEEP SLEEP
	PORG	ON	ON	Hibernate
	AON PD (PRCM)	ON	ON	ON
ANA	SLICER LDO (XO+SLICER)	ON	ON	OFF
	APLL VCO LDO, IOBUF LDO	OFF	OFF	OFF
	MDLL	OFF	OFF	OFF
	SYNTH VCO LDO, DIV_LDO, SDM_LDO	OFF	OFF	OFF
	RX FE, IFA, IFA SAT Detection, ADCs, DC_OFFSET_CORR	OFF	OFF	OFF
	IFA BIAS, ADC BIAS, RX BIAS, IFA LDOs, ADC DIG LDOs	OFF	OFF	OFF
	RF LDO (PA, LNA) , BIAS	OFF	OFF	OFF

	COMPONENTS	IDLE	DEVICE SLEEP (<10ms)	DEVICE DEEP SLEEP
FECSS	FEC SS PD State	ON	ON	RETENTION
		Default FEC PD will be ON state;To force on state write FEC_PWR_REQ_PARAM = 18'h217FF		Before entering Deep Sleep write FEC_PWR_REQ_PARAM = 18'h21FFFAnd for memory retationFEC_PD_MEM_SLEEP_STATE[1:0]
	Cortex M3	CLK GATED	CLK GATED	OFF
		FEC_CM3_CFG_CM3_SYS_RESET_HOLD = 0x7FEC_CM3_CFG_CM3_CLK_GATE = 0x1	OSC_CLK gated in PRCM	Power Off
	FECSS I/C, Peripherals	CLK GATED	CLK GATED	OFF
		IPCFGCLKGATE* = 0x7	OSC_CLK gated in PRCM	Power Off
	DFE	CLK GATED	CLK GATED	OFF
		FEC_RX_ADC_CLKCTL_GATE = 0x7IPCFGCLKGATE0_DFE_CFG = 0x7	OSC_CLK gated in PRCM	Power Off
	RAMPGEN	CLK GATED	CLK GATED	OFF
		FEC_RX_ADC_CLKCTL_GATE = 0x7IPCFGCLKGATE0_RAMPGEN = 0x7	OSC_CLK gated in PRCM	Power Off
HWASS	HWASS PD State	RETENTION	RETENTION	RETENTION
		Default HWA PD will be ON state;To force on state write HWA_PWR_REQ_PARAM = 18'h217FF	Before entering Deep Sleep write HWA_PWR_REQ_PARAM = 18'h21FFFAnd for memory retation HWA_PD_MEM_SLEEP_STATE[3:0]	Before entering Deep Sleep write HWA_PWR_REQ_PARAM = 18'h21FFFAnd for memory retation HWA_PD_MEM_SLEEP_STATE[3:0]
	HWASS I/C, TPTCs, TPCCs	OFF	OFF	OFF
		Power Off	Power Off	Power Off
	HWA 1.2	OFF	OFF	OFF
		Power Off	Power Off	Power Off
TOP SS	FRC	XTAL CLK	CLK GATED	OFF
			OSC_CLK gated in PRCM	Power Off
	PLL_DIG	OFF	OFF	OFF
		PLLDIG_EN_CFG_PLLDIG_EN = 0x0	PLLDIG_EN_CFG_PLLDIG_EN = 0x0	Power Off

	COMPONENTS	IDLE	DEVICE SLEEP (<10ms)	DEVICE DEEP SLEEP
APPSS	APPSS PD State	ON	ON	RETENTION
		Default APP PD will be ON state;To force on state write APP_PWR_REQ_PARAM = 18'h217FF		Before entering Deep Sleep write APP_PWR_REQ_PARAM = 18'h21FFFAnd for memory retation APP_PD_MEM_GRP1_SLEEP_STATE[2:0]APP_PD_MEM_GRP2_SLEEP_STATE[1:0]
	Cortex M4F	WFI, on XTAL CLK	CLK GATED	OFF
		APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	OSC_CLK gated in PRCM	Power Off
	APPSS I/C, Peripherals	on XTAL CLK/2	CLK GATED	OFF
		APP_CPU_CLKCTL_SRCSEL = 0x000APP_CPU_CLKCTL_DIVR = 0x000	OSC_CLK gated in PRCM	Power Off
	CAN FD	OSC_CLKx2	CLK GATED	OFF
		APP_CAN_CLKCTL_SRCSEL = 0x111APP_CAN_CLKCTL_DIVR = 0x000	OSC_CLK gated in PRCM	Power Off

23.6 Wakeup Mechanism from PRCM

The PRCM has two outputs - (i) radar_wakeup_interrupt (ii) radar_devicesleep_wakeup_interrupt.

1. **radar_wakeup_interrupt** - This interrupt is used to wake up the CPU using WIC, when both HCLK and FCLK are gated. Both CM3 and CM4 can be wakeup using this interrupt.
   a. Wakeup sources - GPIO, SYNC_IN, and Counter within PRCM
      i. SYNC_IN is a wakeup source and can be selected using the WAKEUP_IO_MUX_SEL register
      ii. Will choose between GPIO and SYNC_IN
2. **radar_devicesleep_wakeup_interrupt** - The interrupt is used to wake up when APPSS is completely powered down.
   a. Wakeup source logic module generates the wakeup source signals for the RADAR power state FSM. There are four wakeup sources in device sleep counter, UART, SPI and GPIO. This sources can be individually enable or disable by wu_source_en register in the wake registers. The wakeup from UART, SPI or GPIO is the negative edge on these signals and for the sleep counter when its value reached to the sleep_counter_end register value in the wake registers. The final radar_wakeup_source signal is or of all the wakeup sources signals. This signal is used to wakeup device from DEEP_SLEEP or from SLEEP state.

23.7 APPSS WIC Interrupt Map

WIC LINE#	Interrupt Source
0	ESM_HI_IRQ (NMI)
1	ESM_LO_IRQ (INT#1)
2	FECSS_FRAMETIMER_FRAME_START (INT#33)
3	MUXED_FECSS_FRAME_START_OFFSET_INTR_TIME1 (INT#35)
4	FECSS_FRAME_START_OFFSET_INTR_TIME2 (INT#36)
5	FECSS_FRAME_START_OFFSET_INTR_TIME3 (INT#37)
6	FECSS_BURST_START_OFFSET_TIME(INT#38)
7	MUXED_APPSS_RTI1_RTI2_INT_REQ0(INT#43)
8	MUXED_APPSS_RTI1_RTI2_INT_REQ1(INT#44)
9	MUXED_APPSS_RTI1_RTI2_INT_REQ2(INT#45)
10	MUXED_APPSS_RTI1_RTI2_INT_REQ3(INT#46)

WIC LINE#	Interrupt Source
11	APPSS_SPI_IRQ_REQ(INT#14)
12	SPI2_IRQ_REQ (part of INT#28)
13	APPSS_LIN_INT0 (INT#10)
14	APPSS_LIN_INT1 (INT#11)
15	APPSS_MCAN_INT0(INT#21)
16	APPSS_MCAN_INT1(INT#22)
17	APPSS_SCI2_INT0(INT#62)
18	APPSS_SCI2_INT1(INT#63)
19	APPSS_SPI_IRQ_REQ(INT#14)
20	SPI2_IRQ_REQ (part of INT#28)
21	APPSS_LIN_INT0 (INT#10)
22	APPSS_LIN_INT1 (INT#11)
23	APPSS_MCAN_INT0(INT#21)
24	APPSS_MCAN_INT1(INT#22)
25	APPSS_SCI2_INT0(INT#62)
26	APPSS_SCI2_INT1(INT#63)
27	SYNC_IN
28	radar_devicesleep_wakeup_interrupt // from PRCM. This is not same as INT#61 (APPSS_WKUP_INTR).
29 to 31	Reserved

24 Interprocesser Communication

24.1 Introduction

The device provides an inter-processor communication (IPC) mechanism to asynchronously exchange the messages between any two processors.

Simple IPC Mechanism of triggering the initialization, monitoring, low-level RF and calibration functions in FEC controller using register interface between FEC controller and APP host CPU.

The IPC between the processors is Command and Status register based scheme. 32-bit registers are provided which can be used to command the Cortex M3 core and 32-bit Status is configured by the CM3 core which can be read by the Cortex M4F core to get the status of the issued command. Events are generated using SW interrupts cross-connected between the two cores. The IPC can be programmed through DFP APIs.

24.2 Block Diagram

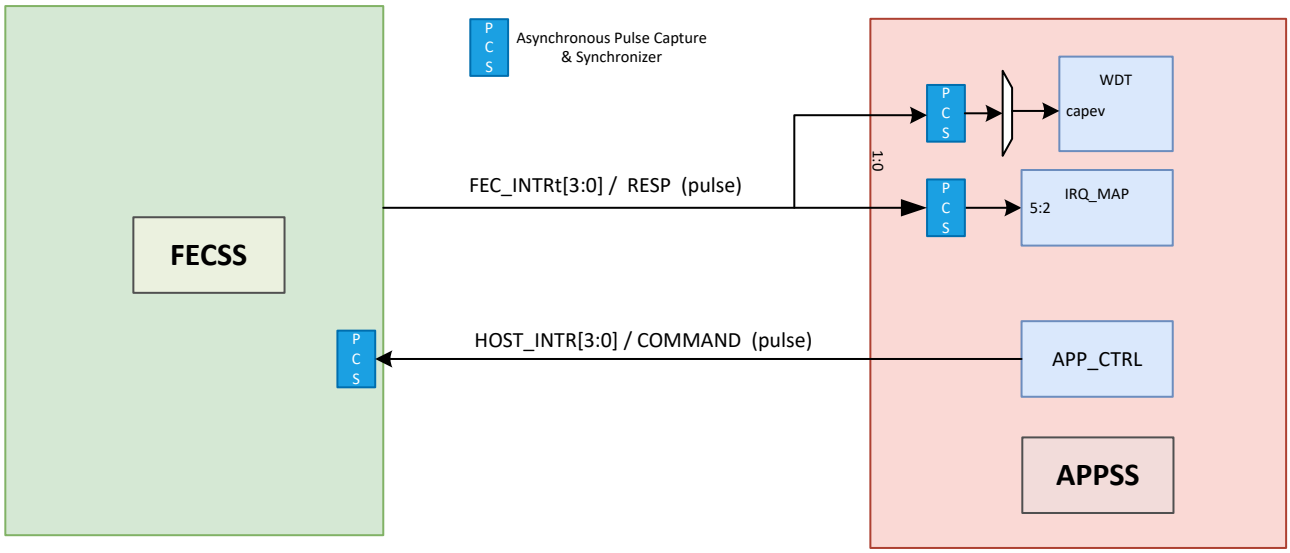


Figure 24-1. Block Diagram

25 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from November 15, 2023 to March 15, 2024 (from Revision * (November 2023) to Revision A (March 2024)) **Page**

- *Global:* Documentation for IWRL6432 applies to IWRL6432AOP ... 8
- Updated GPADC resolution from 10bits to 8bits ... 8
- Updated Bootloader flowchart for warm reset considerations ... 187
- (Clock subsystem) : Updated diagram ... 403
- (Clock subsystem) : Removed "77 to 81GHz spectrum" from introduction ... 403
- (Clock subsystem) : Updated operating frequency range ... 403
- (Transmit Subsystem) : Updated Transmit Subsystem diagram ... 405
- Updated LIN register map and register information ... 1395
- Removed LIN_GLB_INT_CLR ... 1395
- LIN_REGS address corrected to 0x5300_0000 ... 1395
- Replaced LIN_REGS with APP_LIN ... 1395
- Added SCIPIO registers ... 1395

ADVANCE INFORMATION

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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, regulatory 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 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.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265